Models and algorithms for image reconstruction

Hurn, Merrilee Ann

Award date:
1992

Awarding institution:
University of Bath

Models and Algorithms for Image Reconstruction

Submitted by
Merrilee Ann Hurn
for the degree of PhD
of the University of Bath
1992

M.A.Hurn

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

Images are a source of information in a number of application areas, ranging from the medical and biological sciences, to geography and engineering. Unfortunately, it is often the case that in recording the image of interest, various types of degradation can be introduced. The image reconstruction problem is to recover the unobservable true image from the imperfect record.

Chapter 2 of this thesis describes the statistical framework within which the reconstruction problem is formulated. Models are introduced for both the degradation processes, and for the underlying image itself. These models are combined to give a probability distribution on the image given the recorded data. The estimate of the true scene is taken to be the maximiser of this posterior distribution. This estimate will depend upon the forms of the models used. In Chapter 3, various aspects of the prior modelling of the image are considered. Suitable properties are discussed, before concentrating on a model which encourages particular types of smooth behaviour in the image, and promotes the recovery of discontinuities between regions exhibiting different behaviour.

As a result of the high dimension of the problem, there are severe difficulties in actually finding the maximum of the posterior distribution. Two algorithms, one deterministic and the other stochastic, are frequently used, with the pixels of the image generally updated sequentially. In Chapter 4, a review is presented of some existing approaches for incorporating multiple-pixel updates. These approaches vary in their conceptual, and computational, complexity. In Chapter 5, the cascade algorithm, possibly the simplest multiple-site update method, is revised, extended, and implemented. The intention is to improve the performance of both the maximisation algorithms by redirecting their search procedures. An investigation of this revised cascade is presented in Chapter 6.

Acknowledgements

This research was carried out under the supervision of Dr C. Jennison; I am extremely grateful to him for his direction, encouragement, and persistent good humour under provocation.

I would also like to acknowledge the School of Mathematical Sciences of the University of Bath for their excellent computing facilities, without which this work could not have been attempted.

Finally, I would like to thank the Science and Engineering Research Council for their financial support.

Table of Contents

Abstract i
Acknowledgements ii
Table of Contents iii
Notation vi

Chapter 1: Introduction 1
1.1 The problem 1
1.2 The statistical approach and topics considered 2

Chapter 2: The Image Model and Existing Methods 5
2.1 Setting up the model 5
2.1.1 Notation and noise assumptions 5
2.1.2 Bayesian approach to image restoration 7
2.1.3 Smoothed least squares approach 10
2.2 Simulated annealing for minimisation of the energy function 11
2.2.1 Outline of simulated annealing 11
2.2.2 The general Hastings' algorithm 12
2.2.3 The Metropolis algorithm 13
2.2.4 The Gibbs sampler 15
2.3 Temperature schedules for simulated annealing 17
2.3.1 Theoretical results 17
2.3.2 Implementable schedules 19
2.4 Other minimisation techniques 21
2.4.1 Iterated conditional modes 21
2.4.2 Exact MAP for binary images 22

Chapter 3: Choice of Prior 24
3.1 Introduction 24
3.2 Geman and Reynolds' approach 25
3.2.1 Introduction 25
3.2.2 Properties of $\varphi(\,)$ 26
3.3 Higher order models 28
3.4 Parameter selection for Geman and Reynolds' prior 33
3.4.1 Rationale for selection 33
3.4.2 Choice of smoothing parameter λ 35
3.4.3 Choice of scaling parameter Δ 41
3.4.4 Examples 42
3.4.5 Further comments on the parameter selection 50

3.5 Appendix 55
3.5.1 Introduction 55
3.5.2 First order model 56
3.5.3 Second order model 56
3.5.4 Third order model 57

Chapter 4: Multiple-Site Update Methods 63
4.1 The need for multiple site updates 63
4.1.1 Introduction 63
4.1.2 Problems with ICM 63
4.1.3 Problems with simulated annealing 64
4.2 The cascade algorithm 65
4.3 The renormalisation group approach 67
4.4 Swendsen and Wang's algorithm 69
4.4.1 Original proposal 69
4.4.2 Extensions 70
4.5 Multigrid techniques 75
4.5.1 General numerical analysis setting 75
4.5.2 Application to imaging 75
4.6 Some other approaches 77
4.6.1 Pyramid methods 77
4.6.2 Methods in texture analysis 78

Chapter 5: An Extension of Cascade 80
5.1 The conceptual problems with the existing algorithm 80
5.1.1 Inconsistency of prior models 80
5.1.2 Inclusion of blurring 81
5.2 Modifications to the cascade algorithm 83
5.2.1 Modifications to prior contribution 83
5.2.2 Modifications to likelihood contribution 84
5.2.3 Connection between different levels 85
5.2.4 Examples on test scenes 86
5.3 Adaptive cascade 89
5.3.1 Introduction and aims 89
5.3.2 Partitioning scheme 90
5.3.3 Examples 93
5.4 Further implementation extensions 97
5.4.1 A window cascade 97
5.4.2 A diagonal cascade 100
5.5 Some examples with ICM 104

Chapter 6: Simulated Annealing and the Cascade Algorithm 113
6.1 Introduction 113
6.1.1 Outline of chapter 113
6.1.2 An alternative minimisation problem with cascade 115
6.2 Hastings algorithms on the lattice 119
6.2.1 The modified Metropolis algorithm and Gibbs sampler 119
6.2.2 Application of Hajek's result on the lattice 120
6.2.3 A comment on the Geman and Reynolds' truncated algorithm 121
6.2.4 Monitoring the sampling distribution 122
6.3 Annealing on the lattice 124
6.3.1 Hajek's result and the logarithmic schedule 124
6.3.2 Speeding up the logarithmic schedule 131
6.3.3 Annealing with a small number of sweeps 134
6.4 Introducing cascade steps into the schedule 140
6.4.1 Temperature schedules 140
6.4.2 Results 143
6.5 A return to the image problem 147
6.5.1 Temperature schedules with cascade 147
6.5.2 Examples 149
6.6 Conclusions 153

Chapter 7: Conclusions and further work 155

References 158

Notation

Symbol	§	Meaning
$\mathbf{X}$	2.1.1	Vector random variable, the image
S^0	···	The set of all pixels
$\|S^0\|$	···	The number of pixels in S^0, equals dimension of $\mathbf{X}$
$\mathbf{X}_s$	···	The s^{th} component of $\mathbf{X}$
N	···	The number of possible values each $\mathbf{X}_s$ is permitted to take
Ω	···	The space of all possible images on the pixel grid
$\mathbf{Y}$	···	Vector random variable, the record
S	···	The set of pixels for which a record exists
$\mathbf{K}$	···	Matrix representing blurring
B_s	···	The pixels involved in the blurring of pixel s
γ_{s-t}	···	Element $\mathbf{K}_{s,t}$, the weight given to pixel t in blurring of pixel s
η	···	The random Normal noise
σ^2	···	Variance of the η noise components
$\mathbf{x}$	2.1.2	Particular realisation of the image $\mathbf{X}$
$\mathbf{R}$	···	The real numbers
δ_s	···	The Markov random field neighbours of pixel s
c	···	Clique
C	···	Set of all cliques
$\mathbf{X}_{-s}$	···	All components of $\mathbf{X}$ except the s^{th}
$\Phi(\,)$	···	Log prior, or smoothing penalty, for image
$\varphi_c(\,)$	···	Log prior, or smoothing penalty, for clique c
$\mathrm{H}(\mathbf{X})$	2.1.3	The energy function for image $\mathbf{X}$
λ	···	Parameter balancing smoothness and data fidelity in the model
T_k	2.2.1	The annealing temperature at step k
$p_k(\mathbf{x})$	···	The temperature T_k energy distribution
$P(\mathbf{x},\mathbf{x}')$	2.2.2	Probability of arriving in state $\mathbf{x}'$ given started in state $\mathbf{x}$
$q(\mathbf{x},\mathbf{x}')$	···	Proposal distribution of $\mathbf{x}'$ from $\mathbf{x}$
$\alpha(\mathbf{x},\mathbf{x}')$	···	Acceptance distribution of proposal $\mathbf{x}'$ from $\mathbf{x}$
d^*	2.3.1	The maximum over the local minima, of the least climb required to reach another point of lower energy
$D_c(\,)$	3.1	Linear approximation to first order derivative
$\mathrm{I}_{[\,]}$	···	Indicator function
$\mathbf{X}^0$	3.2.1	The unobservable true image
$\mathbf{u}^s$	3.2.2	$\|S^0\|\times 1$ vector taking zero at all coordinates except the s^{th}, where it takes the value u

Δ	$\cdots$	Scaling parameter of the Geman & Reynolds' model
$D_c^i(\,)$	3.3	Linear approximation to i^{th} order derivative
w_c^i	$\cdots$	The weight given in energy to clique c of model order i
$f_s^i(u)$	3.4.1	The change in $\Phi(\,)$ for the i^{th} model due to perturbation $\mathbf{u}^s$
c_i	$\cdots$	Constant bounding the ratio of $f_s^i(u)$ to $\varphi(u)$
$\mathrm{H}(\mathbf{X},\eta)$	3.4.2	The energy of image $\mathbf{X}$ stressing noise η
$\delta_{s,u}\mathrm{H}(\eta)$	$\cdots$	The change in the energy due to a perturbation u at a particular pixel s
Λ	$\cdots$	The event that $\mathbf{X}^0$ is a coordinate-wise minimum
Λ_s	$\cdots$	The event that the conditions for $\mathbf{X}^0$ to be a coordinate-wise minimum are not violated at pixel s
β_s	$\cdots$	The sum of squared blurring weights γ_{t-s} involving pixel s
β	$\cdots$	The maximum over $s\in S^0$ of β_s
$Z_s(\eta)$	$\cdots$	A linear combination of components of η around s
d	$\cdots$	Parameter of the Geman & Reynolds' model
$L^{(m)}$	4.2	The m^{th} level grid in a multilevel system, $m=1, \ldots, M$
$\mathbf{X}^{(m)}$	$\cdots$	The image on the m^{th} level grid
$\mathbf{Y}^{(m)}$	$\cdots$	The aggregated record on the m^{th} level grid
$<s,t>$	$\cdots$	Denotes that pixels s and t are in some two-pixel clique
Q_i	5.3.2	Statistic measuring dissimilarity between pixels in blocks either side of partition site i
ξ	$\cdots$	Maximum number of rows (columns) in pixel strips
ξ_i	$\cdots$	Actual number of rows (columns) in strips for partition site i
Q_i^{ξ}	$\cdots$	Statistic measuring dissimilarity between pixels in strips up to ξ rows (columns) wide either side of partition site i
n	6.1.1	Dimension of lattice
$\{\mathbf{x}\}$	6.1.2	Nodes of the lattice
$V(\,)$	$\cdots$	Function on $n\times n$ lattice
$\mathbf{x}_{\min}$	$\cdots$	Node satisfying $V(\mathbf{x}_{\min})=0$
$p_k(\mathbf{x})$	6.2.1	The temperature T_k distribution on the lattice
$q_n(\mathbf{x},\mathbf{x}')$	$\cdots$	Proposal distribution of node $\mathbf{x}'$ from $\mathbf{x}$ on $n\times n$ lattice
$\alpha_n(\mathbf{x},\mathbf{x}')$	$\cdots$	Acceptance distribution of proposal $\mathbf{x}'$ from $\mathbf{x}$ on $n\times n$ lattice
$\tau_{\mathbf{x}}$	$\cdots$	The lattice neighbours of node $\mathbf{x}$
$\lvert\tau_{\mathbf{x}}\rvert$	$\cdots$	Number of lattice neighbours of node $\mathbf{x}$
K	6.3.3	The number of annealing steps

Chapter 1: Introduction

1.1 The problem

Images are a source of information in many application areas. Satellites measure reflectance from the Earth's surface to provide maps of land use, and pictorial displays are made of the X-ray energy transmitted through soft tissue to detect fractures. The recorded image may be for visual use, a human diagnosis of a possible fracture, or for further numerical processing, possibly calculating the proportion of a particular type of land use. In this work, we will restrict ourselves to the case of directly observed images, such as the two examples mentioned above. We will not consider indirectly observed images, for example as arise in medical scanning techniques such as positron emission tomography. We will also only consider univariate records.

The data may have been recorded continuously across the detecting medium, for example on photographic film (up to grain effects). Alternatively, the data may already be recorded numerically on an array of picture elements, or pixels as they are known. This is the case, for example, in modern optical astronomy where photographic plates have been replaced by banks of recording devices known as charge-coupled-device detectors. The former type of record is usually digitised into the latter for the purposes of computer storage, or processing. Some form of processing may be needed because the detected image will seldom be an entirely accurate representation of the underlying scene of interest. This is a problem which possibly cannot be tackled simply by improving the quality of the recording devices. In some situations, the degradation may be unpredictable. In other situations, an improvement in measurement accuracy may prompt a lowering of input signal; this is particularly true in the medical context where, for example, the lowest possible exposure to X-rays is desirable.

We will consider two sources of degradation, both of which can occur in a number of ways. Firstly, there may be blurring caused by motion of the object of interest relative to the recording device, or as a result of distortion by the lens or by the atmospheric conditions, or for some other reason. Then there may also be sensor noise. This can occur within the recording device itself, for example either as thermal noise affecting electronic equipment, or in low-signal situations as a property of counting incoming packets of information. The final record can be expressed as some function which represents the degradation effects on the true scene, and the way in which they interact: record=f(scene, noise).

The problem to be addressed is the recovery of the true unobservable image, given the observed records. This amounts to attempting to invert the expression given above for the record, without knowing the exact realisations of the noise variables. We will make assumptions about the noise distribution, the form of $f(\ )$, and the nature of the underlying unobservable scene of interest. Since we will be working with records digitised to a pixel grid, our restoration will be restricted to a digitised version of the true scene.

1.2 The statistical approach and topics considered

There are several stages in the image restoration problem at which a statistical input can be made. The first is in attempting to model the sensor noise as a realisation of a random process. Combining this with a model for the blurring, assumed deterministic, using the convolution of these two forms of degradation, we then have a statistical model for the image data, the numerical values recorded on the pixel grid. In Chapter 2, we will discuss the assumptions which will be made about the formation of the record. Our modelling simplifications allow us to write the system in a matrix form reminiscent of a linear model. Unfortunately, a direct inversion to estimate the true scene would be at best ill-conditioned, and at worst impossible if the blurring causes the system to be underdetermined. To tackle this problem, the system can be regularised by the addition of a smoothness constraint. This expresses a belief that the values at pixels which are spatially close in the grid are quite likely to possess a certain degree of smoothness. Statistically, this can be formulated by specifying a locally dependent Markov random field prior model for the underlying scene. A discussion of particular forms of this model type is deferred to Chapter 3.

Combining the prior on the unobservable scene with the likelihood for the records, Bayes theorem can be applied to generate the posterior density of the true scene given the data (up to proportionality). One possible estimate which we will consider for the true scene is the image which maximises this posterior density. An alternative, but equivalent, task is to locate the minimiser of the negative log posterior, or energy function as it is known. This optimisation is far from trivial; the dimension of the energy function for even a small pixel grid makes a direct search over all the possible images computationally infeasible. Chapter 2 concludes with a description of two widely implemented optimisation techniques which are usually applied pixel-by-pixel. One of these is a deterministic steepest descent algorithm, ICM, the other is a stochastic procedure, simulated annealing, which should enable the minimisation to avoid becoming trapped in the numerous local minima of the energy function.

Certain properties of the Markov random field model are desirable in order to comply with prior beliefs in the scene. Satisfying these conditions places restrictions on which classes of potential models can be used as priors. One particular parametric form is described in Chapter 3, in conjunction with three different measures of smoothness. These three levels are selected as being akin to penalising discontinuities in the first, second, and third order derivatives of the image respectively. Processing takes place sequentially in a hierarchy from the first level through to the third. The intention is to recover significant discontinuities in the first, second, and then third order smoothness, whilst attempting to fit up to a quadratic surface to the data. The parameters of the model can be chosen so that, at least with a given probability, discontinuities running horizontally and vertically across the pixel grid will be recovered correctly. It is also possible to extend this idea to the task of recovering edges running diagonally across the grid; we demonstrate this extension.

Algorithmic aspects of the energy minimisation are a topic of much current research interest. Problems exist with both of the pixel-wise iterative algorithms outlined in Chapter 2. The deterministic ICM is only guaranteed to converge to a local minimum of the energy. Asymptotically, the stochastic simulated annealing should perform better, and find the global minimum under certain conditions. However determining these conditions is difficult, and the finite convergence properties are unclear. A better local minimum, or alternatively an increased rate of convergence, could possibly be achieved by allowing more than one pixel to change at each iteration step of the methods. In Chapter 4, we review some existing approaches which incorporate multiple-pixel updates. Some of these methods are intended to improve schemes for sampling from distributions, and so are applicable modifications of simulated annealing, but not of ICM. The remainder are targeted at minimisation, and could be considered as attempts to home in on good regions of the energy function; these could generally be used in conjunction with either of the two algorithms.

In Chapter 5, we consider modifying, and extending one particular multiple-site update method aimed at minimising the energy function, the cascade algorithm. This algorithm attempts to capture various resolution features of the scene, processing at a variety of "pixel" sizes formed by aggregating square blocks of the true pixels. In its original form, cascade was defined only for unblurred scenes, and certain inconsistencies existed between the levels of pixel aggregation used. These problems are dealt with in the redefined version. Cascade can then be regarded as minimising the energy function over increasing subsets of all the possible images as it moves from a coarse to a fine resolution. Experiments incorporating ICM demonstrate that cascade can be badly affected by the fixed

pixel blockings. We propose, and implement, a number of modifications which attempt to find more flexible data-driven aggregation schemes, essentially trying to group together pixels exhibiting similar behaviour. The redefinition of cascade permits this use of non-regular "big pixels", and the processing is carried out much as before.

Any investigation of cascade is hindered by the complexity of the space of all possible images. In an attempt to circumvent this, we draw an analogy between minimising the energy function over the graph defined by the images, and minimising a test function defined over a regular lattice. For sufficiently small lattices, we can monitor many aspects of the behaviour of simulated annealing under various different conditions. This provides some insight into the sensitivity of the algorithm to various factors which might affect its convergence. Experiments can also be carried out on the lattice to simulate the introduction of cascade steps. The results of these experiments suggest ways of combining cascade with simulated annealing in the image case. They also provide an indication of how cascade may be affecting the minimisation process. This work, together with some final cascade experiments with images, is given in Chapter 6.

Chapter 2: The Image Model and Existing Methods

2.1 Setting up the model

2.1.1 Notation and noise assumptions

In this section, we will describe the assumptions which will be made about the data generated in recording the underlying pixellated scene. In doing this, some notation will be introduced to identify variables such as the image, the record and the noise process.

The image is assumed to be a vector random variable **X** defined on a two dimensional region, S^0, which is partitioned into pixels. These pixels are labelled systematically $\{1, 2, \ldots, |S^0|\}$, as shown in Figure 2.1, where $|S^0|$ is the total number of pixels in S^0. At each pixel s, the associated s^{th} element of the vector **X**, denoted $\mathbf{X}_s$, can take a value within a common integer range $\{0, \ldots, N-1\}$. The space of all possible images **X** on the grid is denoted Ω, and contains the $N^{|S^0|}$ combinations of one of N choices at each of $|S^0|$ pixels.

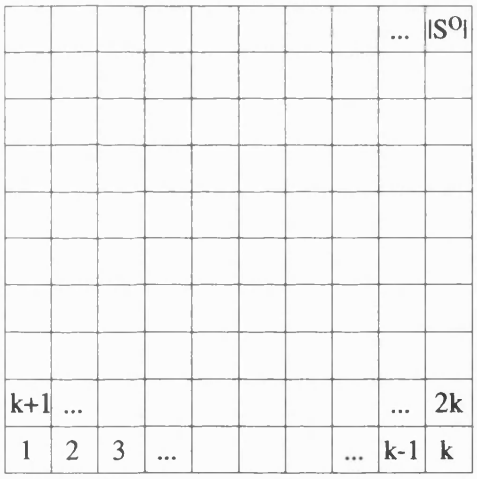


Figure 2.1 The pixel grid.

The image **X** is not observed, rather we record a degraded signal, the continuous random variable **Y** which is defined for some subset S of the pixels. Geman & Geman (1984) propose a model for **Y** which allows for the blurring of the values of **X** before they reach the sensor, and a subsequent convolution of these blurred values with sensor noise. This might be an accurate representation of the process of recording a moving object, for example. The sensor would receive a motion-blurred version of the true scene, and attempt to register this. The recording process itself will introduce degradation, and the final signal will be a convolution of the sensor noise with the blurred values of the underlying scene.

The blurred value at a pixel s is modelled as a weighted average of the value of $\mathbf{X}_s$ with the $\mathbf{X}$ values at certain of the pixels around it in the two dimensional region. Generally this blurring is assumed to be shift-invariant, with blurred values across the scene all generated according to the same pattern of weights. Pixels lying at the edges of the region S^0 may not have the necessary adjacent pixels within S^0 for the blurring process. Records will not exist for these pixels, and for this reason $|S|<|S^0|$ except in the cases either without blurring or where the blurring is assumed to be toroidal.

The blurring weights can be encoded in a shift invariant point-spread matrix $\mathbf{K}$ which acts on the vector $\mathbf{X}$ to give the blurred values. The matrix element $\mathbf{K}_{s,t}$ corresponds to the weight given to $\mathbf{X}_t$ in the blurred value associated with pixel s. If a pixel t is not involved in the blurring of pixel s, then $\mathbf{K}_{s,t}=0$. For notational reasons it would be convenient if the s^{th} element of the record $\mathbf{Y}$ corresponded to the record for the s^{th} pixel. This will not be the case in general because of the dimension difference, unless we introduce dummy records for the pixels in $S^0 \backslash S$. This also requires introducing dummy rows into $\mathbf{K}$. These values are never used, they are coded to an "unknown" value recognised by the data handling programs. This augments $\mathbf{K}$ to an $|S^0|\times|S^0|$ matrix, and $\mathbf{KX}$ and $\mathbf{Y}$ both to $|S^0|\times 1$ vectors. In this way, if $s\in S$, then $(\mathbf{KX})_s$ and $\mathbf{Y}_s$ correspond to the blurred value and record, respectively, at pixel s.

We will introduce the notation B_s, for $s\in S$, to denote the set of pixels $\{t\}$ for which $\mathbf{K}_{s,t}$ is non-zero. These are the pixels whose values are involved in the formation of $(\mathbf{KX})_s$. Since the blurring is shift invariant, and the pixel grid has been labelled as shown in Figure 2.1, $\mathbf{K}_{s,t}$ can be written as γ_{s-t}. Then

$$\begin{aligned}(\mathbf{KX})_s &= \sum_{t:t\in S^0} \mathbf{K}_{s,t}\,\mathbf{X}_t, \quad s\in S \\ &= \sum_{t:t\in B_s} \gamma_{s-t}\,\mathbf{X}_t, \quad s\in S \end{aligned} \tag{2.1}$$

where $\sum_{t:t\in B_s} \gamma_{s-t}=1$ since $(\mathbf{KX})_s$ is a weighted average.

Next we will consider the sensor noise. The assumptions which are made are that the noise is independent of the signal, and also pixel-wise independent, with all the components identically distributed Normally with mean zero and variance σ^2. We will also assume that the convolution of the blurred values with the sensor noise takes the simple additive form. Denoting the noise by η, the observation model may then be expressed

$$\mathbf{Y}_s = (\mathbf{KX})_s + \eta_s, \quad \eta_s \text{ i.i.d. } \sim N(0,\sigma^2), \quad s\in S. \tag{2.2}$$

This is the model which we will use in this work. A discussion of a more general degradation model is given in Geman (1990). To list some of the problems which Equation (2.2) does not address: (i) The effect of the quantisation of a continuous scale into N discrete levels has been ignored. (ii) The underlying image may not be adequately represented by the pixellation scheme (see for example Switzer (1983) for a discussion of mixed pixels in the LANDSAT context, where each pixel can represent a large ground area). (iii) Similarly $\mathbf{K}$ can only approximate the continuous underlying point spread function, which may also not be shift invariant. (iv) As for the sensor noise, this may be neither signal independent, nor Normal, nor pixel-wise independent (some problems of this type arising from astronomical data are described in Molina & Ripley (1989)). (v) The convolution may be more complicated than the additive form we have adopted, and could include an initial transformation of the blurred values.

In the literature concerning blurred images, it seems that $\mathbf{K}$ is usually assumed to be known, either from physical considerations or from estimation on a test image. Ripley (1988, p82) describes an astronomical problem where a general parametric form is available to model the blurring. The two constants of this model are estimated, to allow for variability in the atmospheric conditions and length of exposure, by fitting the parametric model to point sources in the image (isolated stars). We will assume throughout that $\mathbf{K}$, and also σ^2, are known, and their estimation is not discussed here. Two references for estimation of the noise variance are Besag (1986) and Thompson, Brown, Kay & Titterington (1991).

2.1.2 Bayesian approach to image restoration

A Bayesian quantity of interest is the posterior distribution of the scene $\mathbf{X}$ given the data $\mathbf{Y}$. This distribution can be expressed in terms of the likelihood of the data, and a prior for both $\mathbf{X}$ and $\mathbf{Y}$ using Bayes theorem

$$P(\mathbf{X}=\mathbf{x} \mid \mathbf{Y}=\mathbf{y}) = \frac{P(\mathbf{Y}=\mathbf{y} \mid \mathbf{X}=\mathbf{x})\, P(\mathbf{X}=\mathbf{x})}{P(\mathbf{Y}=\mathbf{y})},$$

where lower case letters are used to denote realisations of the random variables.

In the last section, we described the data model. This gives rise to the likelihood for the $|S|$-dimensional real variable $\mathbf{Y}$ given $\mathbf{X}$, using Equation (2.2).

$$\begin{aligned} P(\mathbf{Y}=\mathbf{y} \mid \mathbf{X}=\mathbf{x}) &= P(\mathbf{KX} + \boldsymbol{\eta} = \mathbf{y} \mid \mathbf{X}=\mathbf{x}), && \mathbf{y} \in \mathbf{R}^{|S|},\ \mathbf{x} \in \Omega \\ &= P(\mathbf{Kx} + \boldsymbol{\eta} = \mathbf{y}), && \text{since } \boldsymbol{\eta} \text{ is independent of } \mathbf{X} \\ &= P(\boldsymbol{\eta} = \mathbf{y} - \mathbf{Kx}) \\ &\propto \exp(-(2\sigma^2)^{-1} \| \mathbf{y} - \mathbf{Kx} \|^2) && (2.3) \end{aligned}$$

since the noise process $\boldsymbol{\eta}$ is assumed to be pixel-wise independent, with each component distributed Normally with zero mean and variance σ^2.

In order to apply Bayes theorem and obtain the posterior as a function of $\mathbf{X}$ (up to proportionality), we also require a prior to be specified for the scene. Notice that we are not explicitly interested in $P(\mathbf{Y}=\mathbf{y})$ since it does not depend on $\mathbf{X}$, the variable of interest.

One reasonable prior belief for the scene is that it possesses certain smoothness properties; we might expect pixels which are close in the grid to take values which are similar in some sense. We might also require, for the sake of tractability, that any two pixels separated by more than a certain distance on the grid, are conditionally independent given the values of rest of the scene. Following the work of Geman & Geman (1984), the models generally used for the prior are locally dependent Markov random fields. Before defining a Markov random field, we need to introduce some further notation.

Suppose we define a neighbourhood relation on S^0, with the pixel s a neighbour of pixel t if, and only if, t is a neighbour of s. The intention is to select the neighbourhood structure to be quite local. Denote the set of neighbours of s by δ_s where, by convention, this set is taken to exclude s. A clique c is defined to be either a set of pixels all of whom are neighbours, or a singleton pixel. Let C be the set of all these cliques. Finally, denote the variable consisting of all the components of $\mathbf{X}$, other than the s^{th}, by $\mathbf{X}_{-s}$.

A Markov random field model is then defined by two conditions. The first is that every permitted $\mathbf{x}$ has a non-zero probability, $P(\mathbf{X}=\mathbf{x})>0$, $\forall \mathbf{x} \in \Omega$. The second is that the conditional probability of a pixel s taking a certain value given the rest of the scene will only depend on the values taken by its neighbours δ_s. This condition may be expressed as $P(\mathbf{X}_s=\mathbf{x}_s \,|\, \mathbf{X}_{-s}=\mathbf{x}_{-s}) = P(\mathbf{X}_s=\mathbf{x}_s \,|\, \mathbf{X}_t=\mathbf{x}_t, t \in \delta_s)$. In order to identify the general form of probability density function which satisfies these conditions, we must use the Hammersley-Clifford theorem. This theorem states that a density $P(\mathbf{X}=\mathbf{x})$ is a Markov random field if, and only if, it can be written in the form

$$P(\mathbf{X}=\mathbf{x}) = \exp(-\Phi(\mathbf{x})) / Z, \qquad \mathbf{x} \in \Omega \tag{2.4}$$

where

$$\Phi(\mathbf{x}) = \sum_{c \in C} \varphi_c(\mathbf{x}). \tag{2.5}$$

The term $\varphi_c(\mathbf{x})$ is a function only of those pixels in the clique c, and Z is the normalising constant, sometimes known as the partition function.

The more difficult section of the Hammersley-Clifford theorem is the proof that a Markov random field can always be written in the form of Equations (2.4) and (2.5). A proof is given in Besag (1974). It is far easier, but still interesting, to demonstrate the reverse, that any probability density function satisfying these two equations is indeed a Markov random field:

$$P(\mathbf{X}_s=\mathbf{x}_s \mid \mathbf{X}_{-s}=\mathbf{x}_{-s}) = \frac{P(\mathbf{X}_s=\mathbf{x}_s \cap \mathbf{X}_{-s}=\mathbf{x}_{-s})}{P(\mathbf{X}_{-s}=\mathbf{x}_{-s})}$$

$$= \frac{P(\mathbf{X}=\mathbf{x})}{\sum_{\mathbf{x}'_s} P(\mathbf{X}'_s=\mathbf{x}'_s \cap \mathbf{X}'_{-s}=\mathbf{x}_{-s})}$$

$$= \frac{\exp(-\sum_{c\in C:s\in c}\varphi_c(\mathbf{x}) - \sum_{c\in C:s\notin c}\varphi_c(\mathbf{x}))}{\sum_{\mathbf{x}'_s}\exp(-\sum_{c\in C:s\in c}\varphi_c(\mathbf{x}') - \sum_{c\in C:s\notin c}\varphi_c(\mathbf{x}'))}$$

$$= \frac{\exp(-\sum_{c\in C:s\in c}\varphi_c(\mathbf{x}))}{\sum_{\mathbf{x}'_s}\exp(-\sum_{c\in C:s\in c}\varphi_c(\mathbf{x}'))}.$$

This expression is a function only of those $\mathbf{x}_t$ for which $t\in c$ where c is a clique containing s. From the clique definition, such pixels are the neighbours of s. So provided the $\varphi_c(\mathbf{x})$ take finite values, a distribution defined by Equations (2.4) and (2.5) is indeed a Markov random field.

The choice of a neighbourhood system, and the specification of the $\varphi_c(\,)$ will determine the behaviour of the prior for $\mathbf{X}$. We will discuss these issues in greater detail in Chapter 3. For the moment, we will use the general form given in Equation (2.4) for the prior, and return to the question of the posterior distribution of $\mathbf{X}$ given $\mathbf{Y}$.

We now have sufficient information to specify the posterior as a function of $\mathbf{X}$ by applying Bayes theorem to Equations (2.3) and (2.4),

$$P(\mathbf{X}=\mathbf{x} \mid \mathbf{Y}=\mathbf{y}) \propto \exp(-\Phi(\mathbf{x}) - (2\sigma^2)^{-1}||\mathbf{y}-\mathbf{K}\mathbf{x}||^2). \tag{2.6}$$

The negative of the exponent in this expression, $\Phi(\mathbf{x})+(2\sigma^2)^{-1}||\mathbf{y}-\mathbf{K}\mathbf{x}||^2$, is also known as the energy function. For computational simplicity, it would be an asset if the posterior could also be shown to be a Markov random field. The form of Equation (2.5), which gives the clique breakdown of $\Phi(\,)$, does permit a simplification for the posterior conditional probability of pixel s taking a certain value given the rest of the scene and the record:

$$P(\mathbf{X}_s=\mathbf{x}_s \mid \mathbf{X}_{-s}=\mathbf{x}_{-s}, \mathbf{Y}=\mathbf{y}) = \frac{P(\mathbf{X}_s=\mathbf{x}_s \cap \mathbf{X}_{-s}=\mathbf{x}_{-s} \mid \mathbf{Y}=\mathbf{y})}{\sum_{\mathbf{x}'_s} P(\mathbf{X}_s=\mathbf{x}'_s \cap \mathbf{X}_{-s}=\mathbf{x}_{-s} \mid \mathbf{Y}=\mathbf{y})}$$

$$= \frac{\exp(-\Phi(\mathbf{x}) - (2\sigma^2)^{-1}||\mathbf{y}-\mathbf{K}\mathbf{x}||^2)}{\sum_{\mathbf{x}'_s}\exp(-\Phi(\mathbf{x}') - (2\sigma^2)^{-1}||\mathbf{y}-\mathbf{K}\mathbf{x}'||^2)},$$

where $\mathbf{x}'_{-s}=\mathbf{x}_{-s}$. Then, separating out the terms from the negative exponent of the numerator which involve the value of $\mathbf{X}$ at pixel s,

$$\Phi(\mathbf{x}) + (2\sigma^2)^{-1}||\mathbf{y} - \mathbf{K}\mathbf{x}||^2 = \sum_{c\in C: s\in c} \varphi_c(\mathbf{x}) + (2\sigma^2)^{-1} \sum_{t: s\in B_t} (\mathbf{y}_t - (\mathbf{K}\mathbf{x})_t)^2 +$$

$$\sum_{c\in C: s\notin c} \varphi_c(\mathbf{x}) + (2\sigma^2)^{-1} \sum_{t: s\notin B_t} (\mathbf{y}_t - (\mathbf{K}\mathbf{x})_t)^2, \quad (2.7)$$

where B_t is the set of pixels involved in the blurring of pixel t, as defined in Section 2.1.1. The third and fourth terms in this expansion are independent of $\mathbf{x}_s$, and will cancel in the expression for the conditional distribution. The pixels, other than s, whose values are retained are the prior neighbours of s, and pixels involved jointly with s in forming some blurred value (the latter are the members of the set $\{ \bigcup_{t: s\in B_t} B_t \backslash s \}$). These pixels define the posterior neighbours of s. Notice that this neighbourhood system is symmetric, r is a posterior neighbour of s if, and only if, s is a posterior neighbour of r. This property is inherited directly from the prior if s and r were already prior neighbours. Otherwise, it is clear that

$$r \in \{ \bigcup_{t: s\in B_t} B_t \backslash s \} \iff r\in B_t \text{ for some } t \text{ such that } s\in B_t,\ r\neq s$$

$$\iff s\in B_t \text{ for some } t \text{ such that } r\in B_t,\ s\neq r$$

$$\iff s \in \{ \bigcup_{t: r\in B_t} B_t \backslash r \}.$$

So, by the Hammersley-Clifford theorem, the posterior density is also a Markov random field with a neighbourhood system possibly extended from the prior neighbourhood as a result of the blurring.

Under the Bayes formulation, a possible goal in reconstruction might be to find the *maximum a posteriori* (MAP) estimate of the scene given the data. Maximising the posterior probability as given in Equation (2.6) is equivalent to minimising the negative exponent, or energy function. We shall concentrate on this estimate, although further comments about its suitability will be made in Section 2.4.2.

2.1.3 Smoothed least squares approach

As an alternative to the approach given in the last section, it is possible to regard the recovery of $\mathbf{X}$ from $\mathbf{Y}$ in a non-Bayesian light, as is done for example in Geman & Reynolds (1992). Under the assumptions placed on the record formation, a direct inversion of $\mathbf{KX}=\mathbf{Y}-\boldsymbol{\eta}$ would not be possible in the presence of blurring, since $\mathbf{K}$ is then singular. Furthermore, the least squares approach is ill-posed since the number of records can, at most, only equal the number of pixels. In order to convert the ill-posed least squares problem into one which is better behaved, an additional smoothness penalty can be included. This penalty, $\Phi(\mathbf{X})$, is designed to favour images which are locally smooth in some sense. In order to achieve this, $\Phi(\ )$ is composed of a sum of clique penalties, each one penalising

non-smooth behaviour in the associated clique. As a result, $\Phi(\ )$ can be expressed in the form of Equation (2.5). The resulting energy function H(**X**) is then written

$$\begin{aligned} H(\mathbf{X}) &= \Phi(\mathbf{X}) + \lambda \,||\, \mathbf{Y} - \mathbf{KX} \,||^2 \\ &= \sum_{c \in C} \varphi_c(\mathbf{X}) + \lambda \sum_{s \in S} (\mathbf{Y}_s - (\mathbf{KX})_s)^2. \end{aligned} \tag{2.8}$$

The parameter λ balances the relative weights given to the fidelity to the data and to the smoothness penalty. One possible way to select λ is discussed in Chapter 3. The idea of selection does not contravene $\lambda=(2\sigma^2)^{-1}$ in Equation (2.6), since the relevant scaling difference could be considered absorbed into a rescaled $\Phi(\mathbf{X})$ in Equation (2.8). In this approach, the perspective is on minimising the energy function rather than maximising the posterior probability, although the two are in effect identical.

2.2 Simulated annealing for minimisation of the energy function

2.2.1 Outline of simulated annealing

In the last two sections, we formulated the problem to be solved; we want to find the image **x** which minimises the energy function H(**X**) given in Equation (2.8). This is not a trivial task; if each of the $|S^0|$ pixels can take one of the N values $\{0, 1, \ldots, N-1\}$, then there are $N^{|S^0|}$ possible scenes. Even for small two-colour problems, it is computationally infeasible to attempt the minimisation by an exhaustive search.

Geman & Geman (1984) suggested applying a stochastic relaxation algorithm for the minimisation problem. This algorithm, simulated annealing, was earlier considered by Kirkpatrick, Gelatt & Vecchi (1983), although they did not apply it to the imaging case. The name, and some of the associated terminology, originate from an analogy with the annealing of metals. This is a process of cooling metal sufficiently slowly from a molten state to allow a stable, regular structure to form, so avoiding imperfections setting in the solid. An outline of the simulated annealing algorithm is as follows: We define a temperature schedule, which is a sequence of positive numbers, $\{T_k\}$, for steps $k=1, 2, \ldots$. This sequence satisfies the basic conditions that $T_k \geq T_{k+1}$, $\forall\ k$, and $\lim_{k\to\infty} T_k = 0$. Then at step k, we attempt to draw a sample from the distribution at temperature T_k,

$$p_k(\mathbf{X}=\mathbf{x}) = \frac{\exp(-H(\mathbf{x}) / T_k)}{Z_k}, \qquad \mathbf{x} \in \Omega \tag{2.9}$$

where Z_k is the appropriate normalising constant. This distribution is the posterior density raised to the power $1/T_k$ and suitably renormalised. As the temperature $T_k \to 0$, the distribution $p_k(\mathbf{x})$ increasingly concentrates on those **x** which give low

values of H( ). At T_k=0, the distribution will have positive mass only at the minimising **x**. So, if we are drawing a sample from this distribution, we are selecting the **x** which we wish to find.

In order to draw a sample from a particular member of the sequence of distributions $\{p_k(\mathbf{x})\}$, any one of a family of iterative procedures known as Hastings algorithms can be used. These are initialised with an arbitrary distribution, and should converge to a sample from the desired distribution. The rate of convergence may depend upon the target distribution and the exact form of the sampling algorithm. The general Hastings algorithm, along with two specific examples, is discussed in the following sections.

Complications arise in simulated annealing because we are not just attempting to draw a sample from one particular, fixed temperature distribution $p_k(\mathbf{x})$. Rather, the sampling algorithm is being asked to generate a sample from a distribution which changes after each step as the temperature is lowered. A Hastings algorithm will be used, with the realisation generated at the k^{th} step used to initialise the $k+1^{th}$ step. However, this imposes severe restrictions on the sequences of T_k which can be used. In Section 2.3, some existing theoretical results about temperature schedules will be discussed.

2.2.2 The general Hastings algorithm

Hastings (1970) describes an algorithm to sample from a high-dimensional distribution $\pi(\mathbf{x})$ defined on some countable space Ω. The algorithm is iterative; given some state $\mathbf{x}^{(k)}$ of Ω at step k, a potential new state $\mathbf{x}'$ is generated according to a proposal distribution $q(\mathbf{x}^{(k)},\mathbf{x}')$. With probability $\alpha(\mathbf{x}^{(k)},\mathbf{x}')$ the value $\mathbf{x}'$ is accepted as the next state $\mathbf{x}^{(k+1)}$, otherwise the old value $\mathbf{x}^{(k)}$ is retained. We are, for the moment, considering a homogeneous Markov chain and so we can use the notation $P(\mathbf{x},\mathbf{x}')$ to denote the probability that the new state is $\mathbf{x}'$ given that the old state was **x**. The algorithm then has Markov chain transition function

$$P(\mathbf{x},\mathbf{x}') = \begin{cases} q(\mathbf{x},\mathbf{x}')\ \alpha(\mathbf{x},\mathbf{x}'), & \mathbf{x}' \neq \mathbf{x} \\ 1 - \sum_{\mathbf{x}'' \neq \mathbf{x}} P(\mathbf{x},\mathbf{x}''), & \mathbf{x}' = \mathbf{x}. \end{cases} \tag{2.10}$$

The intention is to choose $P(\mathbf{x},\mathbf{x}')$ so that the target distribution $\pi(\mathbf{x})$ is the equilibrium distribution of $P(\mathbf{x},\mathbf{x}')$. In this way, once the algorithm generates a realisation from $\pi(\mathbf{x})$, the succeeding samples will also be realisations from $\pi(\mathbf{x})$. For an equilibrium distribution to exist, $P(\mathbf{x},\mathbf{x}')$ must be irreducible and aperiodic. For $\pi(\mathbf{x})$ to be the equilibrium distribution, we require that

$$\sum_{\mathbf{x} \in \Omega} P(\mathbf{x},\mathbf{x}')\pi(\mathbf{x}) = \pi(\mathbf{x}'), \quad \forall\ \mathbf{x}' \in \Omega.$$

This condition is known as global balance. If we were to order the states of Ω in some way, and let π be the vector $\{\pi(\mathbf{x})\}$, and $\mathbf{P}$ be the matrix $\{P(\mathbf{x},\mathbf{x}')\}$ corresponding to this ordering, then global balance could be expressed as

$$\mathbf{P}\,\pi = \pi. \tag{2.11}$$

A stronger condition which ensures global balance, but which is easier to confirm, is reversibility or detailed balance,

$$P(\mathbf{x},\mathbf{x}')\;\pi(\mathbf{x}) = P(\mathbf{x}',\mathbf{x})\;\pi(\mathbf{x}'), \quad \forall\; \mathbf{x},\; \mathbf{x}' \in \Omega. \tag{2.12}$$

Under these conditions on $P(\mathbf{x},\mathbf{x}')$, the sequence $\mathbf{x}^{(0)}$, $\mathbf{x}^{(1)}$, ... converges weakly to the distribution $\pi(\mathbf{x})$ from any arbitrary starting value $\mathbf{x}^{(0)}$. The speed of convergence, and the precision of any estimates generated from the samples will depend upon the transition matrix $P(\mathbf{x},\mathbf{x}')$. This is the topic of much current research, see for example Green & Han (1991).

This leaves the issue of choosing $q(\,)$ and $\alpha(\,)$ in such a way as to produce an irreducible and aperiodic $P(\,)$, and to satisfy detailed balance (Equation (2.12)). We will assume that we can select a suitable $q(\mathbf{x},\mathbf{x}')$ so that the chain is irreducible and aperiodic. The different choices for $q(\,)$ will distinguish between the specific examples of Hastings algorithms, two of which will be discussed later. The most common choice then for $\alpha(\mathbf{x},\mathbf{x}')$ to satisfy the conditions is

$$\alpha(\mathbf{x},\mathbf{x}') = \min\left\{ 1\;,\; \frac{q(\mathbf{x}',\mathbf{x})\;\pi(\mathbf{x}')}{q(\mathbf{x},\mathbf{x}')\;\pi(\mathbf{x})} \right\}. \tag{2.13}$$

Peskun (1973) shows that this choice is optimal in terms of minimising the asymptotic variance of any empirical averages calculated from the realisations of the chain. Notice that the target distribution $\pi(\mathbf{x})$ is only required up to proportionality; this avoids the difficult calculation of the partition function.

2.2.3 The Metropolis algorithm

In the imaging case, the sample space Ω is the high-dimensional set of all possible images. Unless there are certain restrictions placed on the proposal $q(\mathbf{x},\mathbf{x}')$, it may be no easier to generate a sample from $q(\,)$ than directly from the intractable $p_k(\mathbf{x})$ (Equation (2.9)). One possible restriction which could be placed on the proposal distribution is that $q(\mathbf{x},\mathbf{x}')$ is zero unless $\mathbf{x}$ and $\mathbf{x}'$ differ at most at only one pixel site. This reduces the number of possible $\mathbf{x}'$ for the potential next state from $N^{|S^0|}$ to $N|S^0|$ (a choice of $|S^0|$ pixels, each of which could then take N values). Algorithms which satisfy this constraint are known as single-site update methods. It is common to employ this type of constraint because of the resulting reduction in the complexity of the problem. It is possible that multiple-site updating might be more effective; some algorithms which tackle this, and attempt to keep the subsequent computational burden in check, are discussed in Chapter 4.

The Metropolis algorithm is the original example of an algorithm of the Hastings type. It was first proposed in the paper by Metropolis, Rosenbluth, Rosenbluth, Teller & Teller (1953). It is a single-site update procedure; given the current image $\mathbf{x}$, a pixel s is selected at random to be updated. An $\mathbf{x}'_s$ is proposed uniformly from among the $N-1$ possible values after the exclusion of the current state $\mathbf{x}_s$. Recalling that $\mathbf{x}_{-s}$ denotes all the components of $\mathbf{x}$ except the s^{th}, $q(\,)$ can then be expressed

$$q(\mathbf{x},\mathbf{x}') = \begin{cases} (\,|S^0|\,(N-1)\,)^{-1}, & \text{if } \mathbf{x}_s \neq \mathbf{x}'_s \text{ and } \mathbf{x}_{-s}=\mathbf{x}'_{-s}, \text{ for some } s \\ 0, & \text{otherwise.} \end{cases} \tag{2.14}$$

Notice that $q(\mathbf{x},\mathbf{x}')$ is symmetric in $\mathbf{x}$ and $\mathbf{x}'$. The acceptance rule $\alpha(\mathbf{x},\mathbf{x}')$ follows from the choice of $q(\,)$ via Equation (2.13), which ensures that detailed balance is maintained.

$$\begin{aligned} \alpha(\mathbf{x},\mathbf{x}') &= \min\left\{ 1\ ,\ \frac{p_k(\mathbf{x}')}{p_k(\mathbf{x})} \right\}, \qquad \text{if } \mathbf{x}_s \neq \mathbf{x}'_s \text{ and } \mathbf{x}_{-s}=\mathbf{x}'_{-s}, \text{ for some } s \\ &= \min\left\{ 1\ ,\ \exp(\,-(\,H(\mathbf{x}')-H(\mathbf{x})\,)\,/\,T_k) \right\}. \end{aligned} \tag{2.15}$$

It is apparent from this $q(\,)$ and $\alpha(\,)$, that it is always possible to move from any image $\mathbf{x}$ to any other image $\mathbf{x}'$, since we have determined that the posterior is a Markov random field, and so $p_k(\mathbf{x})>0$, $\forall \mathbf{x} \in \Omega$. This transition will take at least as many individual updates as there are pixel differences between the two images, and usually more due to the random selection of update sites. However, since at each update there is the possibility of no change, the chain is aperiodic and irreducible.

It is more common, in implementing the algorithm, that pixels are selected for updating in raster order, rather than at random. In this case,

$$q_s(\mathbf{x},\mathbf{x}') = \begin{cases} (N-1)^{-1}, & \text{if } \mathbf{x}_s \neq \mathbf{x}'_s \text{ and } \mathbf{x}_{-s}=\mathbf{x}'_{-s}, \text{ for a given } s \\ 0, & \text{otherwise.} \end{cases} \tag{2.16}$$

The form for the corresponding $\alpha_s(\mathbf{x},\mathbf{x}')$ does not change from Equation (2.15) (in fact, this is the form of the acceptance probability for any symmetric proposal). It is clear that detailed balance is maintained provided we are specifying the same pixel s for the transition both from $\mathbf{x}$ to $\mathbf{x}'$, and back. However, there is a difficulty when we define a single step of a new chain to be the $|S^0|$ sequential steps updating individual pixels 1 through to $|S^0|$. Detailed balance is not satisfied since each step updates the pixels in the same order. The transition matrix $\mathbf{P}$ for the new chain is the product of the individual transition matrices, $\mathbf{P}=\mathbf{P}_1\mathbf{P}_2\ldots\mathbf{P}_{|S^0|}$, where $\mathbf{P}_s$ refers to the update of pixel s. Through the choice of $\alpha_s(\,)$, each individual pixel s step maintains detailed balance with regard to that particular

pixel. However, detailed balance is a strong condition which implies global balance (Equation (2.11)), so $\mathbf{P}_s\pi=\pi$, $\forall s\in S^0$, where π is the vector of $\{p_k(\mathbf{x})\}$.

$$\begin{aligned}
\mathbf{P}\,\pi &= (\,\mathbf{P}_1\,\mathbf{P}_2\,\ldots\,\mathbf{P}_{|S^0|}\,)\,\pi \\
&= (\,\mathbf{P}_1\,\mathbf{P}_2\,\ldots\,\mathbf{P}_{|S^0|-1}\,)\,(\,\mathbf{P}_{|S^0|}\,\pi\,) \\
&= (\,\mathbf{P}_1\,\mathbf{P}_2\,\ldots\,\mathbf{P}_{|S^0|-1}\,)\,\pi, \qquad \text{since } \mathbf{P}_{|S^0|}\pi=\pi \\
&= \pi, \qquad\qquad\qquad\qquad\qquad\qquad \text{since } \mathbf{P}_s\pi=\pi\ \forall\ s\in S^0.
\end{aligned}$$

So, this redefined chain does satisfy global balance, which is the condition required for $p_k(\mathbf{x})$ to be the equilibrium distribution. The chain also has a positive probability of moving from any $\mathbf{x}$ to any other $\mathbf{x}'$ in a single redefined step, and so is aperiodic and irreducible as required.

Equation (2.15) shows that if the proposed change would result in a decrease in the energy, then the new value will always be accepted. On the other hand, if the proposal would result in an increase in energy of size $H(\mathbf{x}')-H(\mathbf{x})$, then the new value will only be accepted with probability $\exp(-(H(\mathbf{x}')-H(\mathbf{x}))/T_k)$. Although the algorithm has been devised working at constant temperature, the role of T_k is apparent here; as the temperature decreases, the probability of accepting an increase in energy also decreases.

There is a computational simplification in calculating $\alpha(\mathbf{x},\mathbf{x}')$ which arises from the choice of a Markov random field model for the image prior. In Section 2.1.2, we showed that in this case the posterior, and therefore $p_k(\mathbf{x})$ as defined by Equation (2.9), are also Markov random fields. Using Equations (2.7) and (2.8), the ratio of probabilities can be written involving only the relevant components of the energy function,

$$\frac{p_k(\mathbf{x}')}{p_k(\mathbf{x})} = \frac{\exp(\,-H(\,\mathbf{x}')\,/\,T_k\,)}{\exp(\,-H(\,\mathbf{x}\,)\,/\,T_k\,)}, \qquad \mathbf{x}'_s\neq\mathbf{x}_s,\ \mathbf{x}'_{-s}=\mathbf{x}_{-s}$$

$$= \exp(-(\sum_{c\in C:s\in c}\{\varphi_c(\mathbf{x}')-\varphi_c(\mathbf{x})\}+\lambda\sum_{t:s\in B_t}\{(\mathbf{y}_t-(\mathbf{K}\mathbf{x}')_t)^2-(\mathbf{y}_t-(\mathbf{K}\mathbf{x})_t)^2\})/T_k).$$

2.2.4 The Gibbs sampler

The Gibbs sampler was introduced by Geman & Geman (1984). Although it was not described in their paper as a specific example of a Hastings algorithm, it is possible to formulate it within that framework.

The algorithm is of the single-site update type. The same comments apply to the Gibbs sampler as applied to the Metropolis algorithm regarding random versus sequential site updating. We shall describe the sequential case, with the proviso that each of the individual pixel updates maintains detailed balance with respect to the appropriate pixel.

At site s in the Gibbs sampler, the image $\mathbf{x}'$ is proposed with $q(\mathbf{x},\mathbf{x}')$ proportional to its probability, $p_k(\mathbf{x}')$, provided that $\mathbf{x}'_{-s}=\mathbf{x}_{-s}$. The constant of proportionality equals the sum of the probabilities of the N images arising from holding $\mathbf{X}_{-s}$ fixed and allowing $\mathbf{X}_s$ to vary. Notice that here, unlike in the Metropolis algorithm, the current image $\mathbf{x}$ may be proposed for $\mathbf{x}'$. Then

$$q(\mathbf{x},\mathbf{x}') = \begin{cases} \dfrac{p_k(\mathbf{x}')}{\sum_{\mathbf{x}''_s} p_k(\mathbf{x}'')}, & \text{if } \mathbf{x}'_{-s}=\mathbf{x}_{-s}, \text{ where } \mathbf{x}''_{-s}=\mathbf{x}_{-s} \\ 0, & \text{otherwise} \end{cases}$$

$$= \begin{cases} \dfrac{\exp(-H(\mathbf{x}')/T_k)}{\sum_{\mathbf{x}''} \exp(-H(\mathbf{x}'')/T_k)}, & \text{if } \mathbf{x}'_{-s}=\mathbf{x}_{-s}, \text{ where } \mathbf{x}''_{-s}=\mathbf{x}_{-s} \\ 0, & \text{otherwise.} \end{cases} \qquad (2.17)$$

Using Equation (2.13), and with the notation that $\mathbf{x}'''_{-s}=\mathbf{x}'_{-s}$ and $\mathbf{x}''_{-s}=\mathbf{x}_{-s}$, the acceptance probability is given by

$$\alpha(\mathbf{x},\mathbf{x}') = \min\left\{ 1 \, , \, \frac{(p_k(\mathbf{x}) / \sum_{\mathbf{x}'''_s} p_k(\mathbf{x}''')) \, p_k(\mathbf{x}')}{(p_k(\mathbf{x}') / \sum_{\mathbf{x}''_s} p_k(\mathbf{x}'')) \, p_k(\mathbf{x})} \right\}, \qquad \text{if } \mathbf{x}'_{-s}=\mathbf{x}_{-s}$$

$$= 1 \quad \text{since } \mathbf{x}''_{-s}=\mathbf{x}_{-s} \text{ and } \mathbf{x}'''_{-s}=\mathbf{x}'_{-s}, \text{ and } \mathbf{x}'_{-s}=\mathbf{x}_{-s}. \qquad (2.18)$$

So, under the Gibbs sampler, the proposal $\mathbf{x}'$ is never rejected. Again, although T_k is held fixed here, the role of the temperature can be deduced. As T_k is reduced, all of the $\exp(-H(\mathbf{x})/T_k)$ terms in Equation (2.17) tend to zero. The denominator sum will be dominated by the largest of the N terms, this being the image with maximum probability among the N choices. As T_k becomes smaller, the probability of proposing the minimising value of $\mathbf{X}_s$, given $\mathbf{X}_{-s}$, will tend to 1.

There is a computational simplification in calculating $q(\mathbf{x},\mathbf{x}')$, which is due to the choice of a Markov random field prior for $\mathbf{X}$. From Equation (2.7),

$$\frac{p_k(\mathbf{x}')}{\sum_{\mathbf{x}''_s} p_k(\mathbf{x}'')} = \frac{p_k(\mathbf{x}'_s \mid \mathbf{x}_{-s}) \, p_k(\mathbf{x}_{-s})}{\sum_{\mathbf{x}''_s} p_k(\mathbf{x}''_s \mid \mathbf{x}'_{-s}) \, p_k(\mathbf{x}'_{-s})}, \qquad \mathbf{x}'_{-s}=\mathbf{x}_{-s} \text{ and } \mathbf{x}''_{-s}=\mathbf{x}'_{-s}$$

$$= \frac{\exp\left(-\left(\sum_{c\in C: s\in c} \varphi_c(\mathbf{x}') + \lambda \sum_{t: s\in B_t} (\mathbf{y}_t-(\mathbf{K}\mathbf{x}')_t)^2\right) / T_k\right)}{\sum_{\mathbf{x}''} \exp\left(-\left(\sum_{c\in C: s\in c} \varphi_c(\mathbf{x}'') + \lambda \sum_{t: s\in B_t} (\mathbf{y}_t-(\mathbf{K}\mathbf{x}'')_t)^2\right) / T_k\right)}.$$

2.3 Temperature schedules for simulated annealing

2.3.1 Theoretical results

We have now given the details of two different iterative algorithms for generating a sample from the fixed temperature distribution $p_k(\mathbf{x})$, defined in Equation (2.9). As outlined in Section 2.2.1, simulated annealing attempts to find the minimiser of the energy function, H(**X**), by applying algorithms such as these two, to the case where the distribution is altering at each step as the temperature T_k is decreased. A rigorous analysis of the convergence of the process is complicated by the inhomogeneity of the generated Markov chain, but some theoretical results are given in Geman & Geman (1984), Hajek (1988), and various other papers. All of these results suggest that unless the temperature is lowered very slowly, simulated annealing may not converge to the minimising distribution. This might also be the intuitive view; if at temperature T_k, we are close to generating a sample from the equilibrium distribution $p_k(\mathbf{x})$ then, if at temperature T_{k+1} the distribution $p_{k+1}(\mathbf{x})$ is not greatly different, we may stand a better chance of also being close to generating a sample from this new equilibrium distribution.

Geman & Geman (1984) provide an analysis for simulated annealing based on the Gibbs sampler (Section 2.2.4). Under the condition that each pixel is considered for updating at finite intervals, they show that the samples $\{\mathbf{X}^{(k)}\}$ will converge in distribution to the global minimiser as the number of steps $k \to \infty$, provided that the schedule satisfies

$$T_k \geq \frac{c}{\log(1+k)} \quad \text{for } k=1, 2, \ldots \tag{2.19}$$

where the constant c equals the number of pixels, $|S^0|$, $\times$ the maximum absolute energy difference between any two $\mathbf{x}$ differing at just one pixel. In the systematic raster scan implementation which we will use, k is taken to be the number of complete sweeps of the image, rather than the number of individual pixel visits. Under this interpretation, the temperature T_k is maintained for the $|S^0|$ pixel visits, then $k \to k+1$ and $T_k \to T_{k+1}$. The use of raster scan updating of pixels was justified in Section 2.2.3, and clearly satisfies the Gemans' requirement of finite interval updating of all pixels.

Hajek (1988) considers Metropolis-based simulated annealing (Section 2.2.3). He provides a necessary and sufficient condition for $\{\mathbf{X}^{(k)}\}$ to converge in probability to the global minimiser as $k \to \infty$. In order to state Hajek's condition, we first need to define the depth of a local minimum of the energy function: Suppose we are at some local minimiser $\mathbf{x}$, and want to reach any other image possessing lower energy, $\mathbf{x}'$ say. In accordance with the proposal mechanism, there will be some potential route from $\mathbf{x}$ to $\mathbf{x}'$ consisting of higher energy

intervening images, $\mathbf{x}\rightarrow\mathbf{x}^{(1)}\rightarrow\mathbf{x}^{(2)}\rightarrow\ldots\rightarrow\mathbf{x}'$, where $H(\mathbf{x})\leq H(\mathbf{x}^{(i)})\ \forall\ i$, and $H(\mathbf{x}')<H(\mathbf{x})$. In the case of single-site updating, the transition between each pair of these images involves altering exactly one pixel. The net increase in energy required for the route is $\max_i \{H(\mathbf{x}^{(i)})\}-H(\mathbf{x})$. The depth of the local minimum $\mathbf{x}$ is then defined to be this increase minimised over the various valid routes $\mathbf{x}\rightarrow\ldots\rightarrow\mathbf{x}'$ to all the possible lower energy $\mathbf{x}'$.

Now using the definition of the depth of a local minimum, we can state Hajek's condition. If $T_1, T_2, \ldots$ is a sequence of strictly positive numbers such that $T_1\geq T_2\geq\ldots$ and $\lim_{k\rightarrow\infty} T_k=0$, then Hajek shows that a necessary and sufficient condition for $\lim_{k\rightarrow\infty} P(\mathbf{X}^{(k)}=\text{global minimiser})=1$ is that

$$\sum_{k=1}^{\infty} \exp(-d^* / T_k) = +\infty \qquad (2.20)$$

$$\text{where}\quad d^* = \max_{\text{local minima } \mathbf{x}} \{\text{depth of } \mathbf{x}\}.$$

This condition is certainly satisfied when T_k is of the logarithmic form given by Equation (2.19), provided that the constant $c\geq d^*$; that is, the constant c must be sufficiently large to exceed the net energy climb required to escape from all local minima of the energy function.

These two conditions, Equation (2.19) applying to the Gibbs sampler, and Equation (2.20) applying to the Metropolis algorithm, show that the simulated annealing technique can find the MAP estimate, at least asymptotically. In practise, they do not provide useful advice for choosing a schedule to implement. Both results require constants involving knowledge of the entire function H(X). Knowledge of the entire energy function would make simulated annealing redundant. Worse, Hajek's result shows that for convergence, simulated annealing requires an infinite number of sweeps at positive T_k. Any implementable version of simulated annealing will have to violate these theoretical conditions. In Section 6.3, we will demonstrate Hajek's theoretically correct schedule, and some deviations from it, when the function to be minimised is defined on an $n\times n$ lattice where we can find d^* and monitor the entire sampling distribution. The convergence is seen to be very slow.

Geman and Geman demonstrate several simulated annealing reconstructions of degraded images. They follow schedules of the form of Equation (2.19) taking c between 3 and 4, and truncating the schedule after 300 to 1000 complete raster scans. Hajek does not implement any schedules. In the next section, we will cite some experimental results for different schedules, and also give some justification for schedules other than the logarithmic when the number of sweeps is finite.

2.3.2 Implementable schedules

Stander (1992) provides an extensive comparison of the performance of finite-sweep annealing schedules, chosen from different families of parametric curves, for restoring images. Among the schedules he considers are logarithmics truncated after a certain number of sweeps, and curves which decrease, in the same number of sweeps, and from the same starting values to values close to zero, either linearly or geometrically. Simulated annealing is a stochastic algorithm, and so the exact outcome of the updates will depend on the random variables generated in the implementation. The performance of each schedule is assessed by the mean, and variance, of the energy of 100 reconstructions produced using different seeds for the pseudo-random number generator. The conclusion from his experiments is that the most important features of the schedule are the initial and final temperature values, rather than the choice of temperatures in between. A low starting and finishing temperature were both recommended as beneficial; in particular, he suggests finishing all schedules with zero temperature annealing. The effect of zero temperature annealing is either to permit only those updates which lead to a reduction in energy (for the Metropolis algorithm), or to maximise the possible reduction in energy (for the Gibbs sampler). This will not greatly affect the performance of schedules which already finish with low values, such as the linearly and geometrically decreasing curves. However, for a logarithmic schedule with its very slow rate of decrease, the additional zero-temperature annealing produces a great improvement. In light of the computational savings, and the comparatively poor performance of the truncated logarithmic schedule, Stander concludes that a schedule decreasing linearly from some small value to zero provides a good compromise performance in the finite-sweep case.

We have not conducted any additional experiments on images, although in Section 6.3, we will demonstrate the effect of various different schedules on the entire sampling distribution for an alternative test problem. In this section, we will reconsider Hajek's result (Equation (2.20)), bearing in mind that we can only implement a finite number of sweeps. Hajek's result states that in order to attain the desired convergence, the sequence $\{T_k\}$ must satisfy

$$\sum_{k=1}^{\infty} \exp(-d^* / T_k) = +\infty, \quad \text{where } T_1 \geq T_2 \geq \ldots \text{ and } \lim_{k\to\infty} T_k = 0.$$

From the summation, we can see that there cannot be a finite-sweep schedule which will converge as required. In practise, a truncated logarithmic schedule is often used on the grounds that the slow cooling is intuitively acceptable, and also because we know that this schedule could be extended, with more sweeps, eventually to form a correct schedule, provided that the initial temperature is sufficiently high. Denote the limiting logarithmic schedule by $\{\tilde{T}_k\}$,

$$\tilde{T}_k = \frac{d^*}{\log(\, 1+k\,)} \qquad \text{for } k = 1,\ 2,\ \ldots$$

Suppose we consider the schedule $\{T_k^{(1)}\}$ which begins at a lower initial temperature by effectively starting a finite number of steps n into schedule $\tilde{T}_k$,

$$T_k^{(1)} = \frac{d^*}{\log(\, n+k\,)} \qquad \text{for some finite } n \in \mathbf{N}^+.$$

Then $$\sum_{k=1}^{\infty} \exp(\, -d^* \,/\, T_k^{(1)}\,) = \sum_{k=1}^{\infty} \exp(\, -d^* \,/\, \tilde{T}_k\,) - \sum_{k=1}^{n-1} \exp(\, -d^* \,/\, \tilde{T}_k\,)$$

$$= +\infty \text{ since } n \text{ is finite.}$$

By Hajek's theorem, following this modified schedule would also result in the desired asymptotic properties. We could then consider speeding up the cooling by only selecting every m^{th} value of $T_k^{(1)}$, where m is again a finite positive integer,

$$T_k^{(2)} = \frac{d^*}{\log(\, n+1+(k-1)m\,)}, \qquad \text{for some finite } n,\ m \in \mathbf{N}^+.$$

Notice that $T_k^{(2)} \geq T_i^{(1)}, \qquad k=1,\ 2,\ \ldots,\ i \in \{m(k-1)+1,\ \ldots,\ mk\}$

$$\Rightarrow \quad \exp(\, \frac{-d^*}{T_k^{(2)}}\,) \geq \exp(\, \frac{-d^*}{T_i^{(1)}}\,), \qquad k=1,\ 2,\ \ldots,\ i \in \{m(k-1)+1,\ \ldots,\ mk\}$$

$$\Rightarrow \quad \exp(\, \frac{-d^*}{T_k^{(2)}}\,) \geq \frac{1}{m} \sum_{i=m(k-1)+1}^{mk} \exp(\, \frac{-d^*}{T_i^{(1)}}\,), \qquad k=1,\ 2,\ \ldots$$

$$\text{so} \quad \sum_{k=1}^{\infty} \exp(\, \frac{-d^*}{T_k^{(2)}}\,) \geq \sum_{k=1}^{\infty} \frac{1}{m} \sum_{i=m(k-1)+1}^{mk} \exp(\, \frac{-d^*}{T_i^{(1)}}\,)$$

$$\geq \frac{1}{m} \sum_{i=1}^{\infty} \exp(\, \frac{-d^*}{T_i^{(1)}}\,)$$

$$= +\infty, \text{ since } m \text{ is finite.}$$

Again by Hajek's theorem, this late-start, accelerated-cooling logarithmic schedule will also give the desired asymptotic convergence. It seems that there is slightly more freedom within the logarithmic family of schedules than is generally used. In Section 6.3, we will compare the performance of some of these modified logarithmic schedules over a large number of sweeps.

There is a second point to note from these late-start, accelerated logarithmic schedules. We know that any schedule which monotonically decreases to zero, and lies entirely above, or on, $\{T_k^{(2)}\}$ will also be a suitable schedule for Hajek's result to apply. So, suppose we have any monotonically decreasing schedule starting from any positive initial temperature, and terminating after a finite number of steps

K, at some strictly positive final temperature. It is possible to find a value of n and m so that there is a corresponding $\{T_k^{(2)}\}$ which lies beneath this truncated schedule for $k=1, \ldots, K$. Although the new schedule is truncated after K steps, it could be assumed that had we continued, we would then have followed these $\{T_k^{(2)}\}$ values. So, this new arbitrary schedule could have been extended to form a schedule which would guarantee asymptotic convergence. This statement does not imply anything about the relative merits, over a finite number of sweeps, of the logarithmic and other schedules, since the important behaviour for Hajek's theorem is in the tail as $k\to\infty$. It does provide some justification for selecting any monotonically decreasing schedule without, in practise, the need to follow the logarithmic, or to have any knowledge of d^*.

2.4 Other minimisation techniques

2.4.1 Iterated conditional modes

Simulated annealing is a very computationally demanding technique; on each sweep, every pixel site is visited to draw a sample for the site's potential new value. Depending on the size of the space of possible $\mathbf{x}$, there are unlikely to be fewer than one hundred sweeps. Even after all this work, we have no guarantee of finding the global minimiser of the energy function.

Besag (1986) describes a simple iterative method which he calls Iterated Conditional Modes, or ICM. At each pixel s, a new value $\mathbf{x}_s$ is chosen to maximise the conditional probability of $\mathbf{X}_s$ given the current values of $\mathbf{X}_{-s}$ and the record $\mathbf{y}$. Since $p(\mathbf{x}|\mathbf{y})=p(\mathbf{x}_s|\mathbf{x}_{-s},\mathbf{y})p(\mathbf{x}_{-s}|\mathbf{y})$, each successive $\mathbf{x}$ can only maintain, or increase, the probability of the current reconstruction. This guarantees convergence to a local minimum of the energy function, at least. Notice that the procedure is a strictly downhill, steepest descent algorithm; it cannot escape from local minima, nor is there any assurance that the minimum it finds will be a reasonable one. ICM is equivalent to using the Gibbs sampler at temperature zero.

ICM is computationally fast and cheap. Convergence generally occurs in under fifteen complete sweeps of the image, and certainly in fewer sweeps than would be recommended for simulated annealing. There is also a computationally convenient form for the conditional probabilities, from Equations (2.6) and (2.7),

$$p(\mathbf{x}_s \mid \mathbf{x}_{-s}, \mathbf{y}) = \frac{\exp\left(-\left(\sum_{c\in C:s\in c}\varphi_c(\mathbf{x}) + \lambda \sum_{t:s\in B_t}(\mathbf{y}_t-(\mathbf{K}\mathbf{x})_t)^2\right)\right)}{\sum_{\mathbf{x}'}\exp\left(-\left(\sum_{c\in C:s\in c}\varphi_c(\mathbf{x}') + \lambda \sum_{t:s\in B_t}(\mathbf{y}_t-(\mathbf{K}\mathbf{x}')_t)^2\right)\right)}, \quad \mathbf{x}'_{-s}=\mathbf{x}_{-s}$$

$$\propto \exp\left(-\left(\sum_{c\in C:s\in c}\varphi_c(\mathbf{x}) + \lambda \sum_{t:s\in B_t}(\mathbf{y}_t-(\mathbf{K}\mathbf{x})_t)^2\right)\right).$$

Finding the $\mathbf{x}_s$ which maximises this conditional probability is equivalent to finding the $\mathbf{x}_s$ which minimises the term in the negative exponent; the denominator need not be calculated. There is the additional saving that the minimisation need not be considered for a particular pixel if in the lapse since its last update, none of its posterior Markov random field neighbours have changed in value.

The drawback to ICM is the quality of the final reconstruction. Without the capacity to escape from local minima, ICM rarely finds reconstructions with energy as low as those produced by finite-sweep simulated annealing. It is heavily dependent on the image used to initialise the algorithm, and is particularly badly affected by high degradation levels.

2.4.2 Exact MAP for binary images

In the specific unblurred case where the $\{\mathbf{X}_s\}$ can only take the values 0 or 1, and the prior is of a certain type known as an Ising model (which will be defined in Section 3.1), Greig, Porteous & Seheult (1989) show that it is possible to find the MAP estimate exactly. To do this, they use a variant of the Ford-Fulkerson algorithm for finding the maximum flow through a capacitated network. We will not describe the method here, except to mention that suggestions for computational improvements can be found in Jubb (1989).

There are two issues raised by the exact evaluation of the MAP estimate. The first is that, in this particular class of restoration problems, we can assess the absolute performance of the finite-sweep simulated annealing and ICM algorithms. Greig, Porteous and Seheult present a comparison of various reconstructions using the three methods. The results are presented both visually, and numerically in terms of the percentage of pixels misclassified and the deviation of the reconstruction energy above that of the MAP estimate. The performance of single-site update simulated annealing and ICM as minimisers of the energy function turns out to be quite poor in most cases. In all the examples shown, simulated annealing attains a lower energy value than ICM, with the more successful of its two schedules, logarithmic or geometric, depending on a scaling parameter of the model $\Phi(\mathbf{X})$.

The second, possibly more important issue is whether MAP estimation is actually a suitable image estimate to use. In some of the examples, the MAP estimate is clearly oversmoothed, and has the highest number of misclassified pixels. Although they have higher energies, the simulated annealing and ICM solutions are more acceptable visually, with lower pixel misclassification rates. In a decision theory setting, MAP estimation corresponds to using a loss function which is 0 for the selecting the true image, and 1 for selecting any other image, no matter how different. The MAP estimate may be presenting too global a view of the image at the expense of local features. An alternative formulation is to

consider a loss function where each misclassified pixel incurs a penalty of 1. The corresponding estimate is MPM (marginal posterior modes), in which the estimate for pixel s is the $\mathbf{x}_s$ which maximises the posterior probability $P(\mathbf{X}_s=\mathbf{x}_s|\mathbf{y})$. This is approximated by simulating from the posterior, at temperature 1, for an arbitrary number of sweeps, noting the frequency with which each of the N values occurs for each pixel. The restoration at pixel s is taken to be the most frequently occurring value of $\mathbf{x}_s$. Marroquin, Mitter & Poggio (1987) strongly propose using MPM in preference to MAP, although they claim that the two are very similar in the high signal-to-noise case. The issue is also discussed by Besag (1989).

Ripley (1988, p101) presents two restorations of a binary image, both with the same pixel misclassification rate. One restoration is far more visually acceptable than the other because of the different spatial clusterings of the misclassified pixels across the scene. The choice of estimate, MAP or MPM, must be influenced by the end use of the restoration. If we wish to use the restoration in calculating some statistic involving the number of pixels of each particular value, for example calculating land use from satellite data, then MPM may be the better choice. However, if the restoration is to be viewed in more global terms, either as an image per se, or possibly as a starting point for further processing, such as some form of shape analysis perhaps, then MAP estimation might be preferred.

There is the additional argument that if the energy function is highly multi-modal, it may be insufficient to present a single point estimate at all. It should be possible to use sampling techniques to produce a range of images representative of the posterior distribution. Again, the end use will influence this decision. We have chosen to present a point estimate, and given this choice, we will concentrate on the MAP estimate. In terms of the modelling of the scene, MPM and MAP will both highlight extreme behaviour, and may be useful indicators of suitable modifications. The problems of calculating the MAP estimate encompass some of those of sampling; in some ways, the maximisation is the harder problem. The techniques used, Hastings algorithms and simulated annealing, are also applicable to problems outside imaging. Any algorithmic improvements may carry over into fields in which MAP estimation is a more clear-cut choice. Obviously, there is no one definitive answer in this area.

Chapter 3: Choice of prior

3.1 Introduction

In Section 2.1, we discussed the modelling assumptions which lead to the energy function given in Equation (2.8). We denoted the smoothness penalty, or log prior, for an image $\mathbf{X}$ by a general form, $\Phi(\mathbf{X})$. In this chapter, we will consider the $\Phi(\,)$ function in more detail, stating some desirable properties, and then discussing one particular prior in depth. An approach to parameter selection for this model will be described, and demonstrated with several reconstructions.

The $\Phi(\,)$ function can be considered either as defining a Markov random field (Equations (2.4) and (2.5)), or as a smoothness function penalising local discontinuities in the image. In either interpretation, it can be decomposed into the sum of local contributions from the individual cliques. We will make an assumption of homogeneity in that we will penalise discontinuities in the same way across the image. Then, $\Phi(\,)$ could be expressed in the following way:

$$\begin{aligned} \Phi(\mathbf{X}) &= \sum_{c \in C} \varphi_c(\mathbf{X}) \\ &= \sum_{c \in C} \varphi(D_c(\mathbf{X})) \end{aligned} \tag{3.1}$$

where $D_c(\mathbf{X})$ is some function of the components of $\mathbf{X}$ in clique c, and is considered to be a measure of smoothness. One natural choice for $D_c(\,)$ is as a linear approximation to the first order derivative of the scene,

$$D_c(\mathbf{X}) = \mathbf{X}_i - \mathbf{X}_j, \quad \text{for } i, j \in c, \tag{3.2}$$

where the cliques are pairs of adjacent pixels.

The properties of $\Phi(\,)$ will depend on the properties of $\varphi(\,)$, and the choice of $D_c(\mathbf{X})$. In turn, these attributes should reflect the nature of the image to be restored. Suppose the image consists of a number of unordered colours. For example, $\mathbf{X}$ could be a binary variable indicating presence/absence of some quantity, or we might have a labelling problem where each colour corresponds to a particular type of pixel. In these situations, we might want to use $D_c(\,)$ of the form of Equation (3.2), and impose a constant penalty for any pixel-pair discrepancy, independent of its size. An example of this type is the Ising model,

$$\varphi(D_c(\mathbf{X})) = \beta\, I_{[D_c(\mathbf{X}) \neq 0]} \tag{3.3}$$

where $I_{[\,]}$ is the indicator function taking the values 1 if the event [ ] is true, and 0 otherwise. In this case, when adjacent pixels take different values we impose a penalty of β, otherwise for matching pixel pairs, there is no penalty.

We intend to concentrate on a different class of images, known as grey-level problems, in which an ordering does exist between the possible pixel values. Here, for example, the pixel levels might represent the intensity of light reflected from a surface, or the amount of X-ray transmission through an object. In these situations, it seems reasonable that a small value of $D_c(\ )$ should receive a different penalty than a large value since they may indicate different types of discontinuity, within region variability and between region differences. As we are intending to penalise lack of smoothness in the image, $\varphi(D_c(\mathbf{X}))$ is taken to be even and increasing in $|D_c(\mathbf{X})|$, with $\varphi(0)=0$. These conditions are generally accepted for grey-level scenes, see for example Besag (1989) where the priors $\varphi(u)=u^2$, $\varphi(u)=|u|$, and $\varphi(u)=\log(\cosh(u))$, all of which satisfy these conditions, are discussed. In Sections 3.2.2 and 3.3, we will discuss some possible additional properties for a grey-level $\varphi(\ )$, and alternative choices for $D_c(\ )$, respectively.

3.2 Geman and Reynolds' approach

3.2.1 Introduction

Geman & Reynolds (1992) discuss properties suitable for priors used to model grey-level images. They identify the behaviour of $\varphi(\ )$ which may be beneficial for both the removal of blur from the record, and the recovery of discontinuities in the image. The discontinuities in which they are interested are sharp transitions in either the pixel values, their first order or their second order derivatives. This extends the idea of smoothness beyond images consisting of regions of constant grey-level; it should lead to the recovery of the planar and quadric surfaces which might realistically be expected to exist in the true scene. These aims are approached by specifying a particular family of $\varphi(\ )$, and by suitably extending the definition of the $D_c(\ )$ expression; we will discuss these choices in greater detail in the next two sections.

The paper then suggests, for the purposes of model validation, that it should be possible to define the energy function so that the true image, $\mathbf{X}^0$ say, is actually the sought-after minimiser. This is an ambitious task, however the paper succeeds in defining an energy such that certain prototype $\mathbf{X}^0$, namely horizontal and vertical step edges, are at least local minima of the energy. In order to do this, the parameters of the model are chosen as functions of the noise and the blurring coefficients, rather than being estimated. Geman and Reynolds consider the model with a four neighbour system; in Section 3.4, we derive the equivalent parameter selection for an eight neighbour model, extending the class of prototype images to include diagonal step edges.

3.2.2 Properties of $\varphi(\,)$

We have already identified certain suitable properties for potential grey-level priors $\varphi(u)$. These properties were that $\varphi(u)$ was (i) even, (ii) increasing in $|u|$, and (iii) $\varphi(0)=0$. Satisfying these three conditions still leaves a fairly wide class of possible priors. Geman and Reynolds consider the restoration goals, namely the removal of blurring, and the recovery of both smooth regions and the transitions between these regions. The role that $\varphi(\,)$ plays in these objectives then generates some further useful conditions on $\varphi(\,)$.

First we will consider the removal of the blurring. The data is a blurred, corrupted version of $\mathbf{X}^0$, and so the prior should be able to counteract any small perturbation of a reconstruction towards the data. Suppose we consider some small perturbation of order u at a pixel s. In vector notation, this can be denoted by $\mathbf{u}^s$, where $\mathbf{u}^s$ is the $|S^0|\times 1$ vector taking the value zero at all components except the s^{th}, where it takes the value u. In the expression for the energy, Equation (2.8), the change in the data component can be expanded using Equation (2.1)

$$\sum_{t:s\in B_t}\{(\mathbf{Y}_t-(\mathbf{K}(\mathbf{X}+\mathbf{u}^s))_t)^2-(\mathbf{Y}_t-(\mathbf{KX})_t)^2\} = u(-2\sum_{t:s\in B_t}\gamma_{t-s}(\mathbf{Y}_t-(\mathbf{KX})_t))+u^2\sum_{t:s\in B_t}\gamma_{t-s}^2.$$

So a reduction of order u in the data contribution to the energy can be achieved by the perturbation. To determine the effect on the prior, we can Taylor expand the relevant prior contributions about $\mathbf{X}$,

$$\sum_{c\in C:s\in c}\{\varphi_c(\mathbf{X}+\mathbf{u}^s)-\varphi_c(\mathbf{X})\} = \sum_{c\in C:s\in c}\{u\,\varphi'_c(\mathbf{X}) + \frac{u^2}{2}\varphi''_c(\mathbf{X})+\ldots\}.$$

It can be seen that the change in the prior contribution can only be guaranteed to be of order u if $\varphi'_c(\mathbf{X})\neq 0\ \forall\mathbf{X}$. Therefore, in order that the blurring can be tackled effectively, we have a fourth condition on $\varphi(u)$, (iv) $\varphi'(u)\neq 0,\ \forall u$. Since $\varphi(\,)$ is even and increasing in $|u|$, this implies that the prior must be non-differentiable at the origin, and strictly increasing in $|u|$.

Next we will consider the recovery of transitions between distinct regions. The condition that $\varphi(u)$ is increasing in $|u|$ expresses our belief that the scene consists of smooth regions. However, it also seems reasonable to expect that these regions may be separated by large discontinuities in the smoothness. So there are two types of discontinuities to be tackled. The first is due to degradation within otherwise smooth regions; the prior should be able to deal with these and reconstruct a smooth region. This suggests a rapid growth of $\varphi(u)$ for small $|u|$. The second type of discontinuities are the genuine transitions between different regions. It might be hoped that these are larger in magnitude than the first type. In order not to penalise these boundaries to the extent that they will not be recovered, the prior should have a finite asymptotic limit as $u\to\infty$. In this way,

beyond a certain value of $D_c(\mathbf{X})$, the smoothness penalty in the energy will be approximately constant. The data fidelity term should then allow the discontinuity to be recovered. This gives a fifth condition on $\varphi(u)$, (v) $\lim_{|u|\to\infty} \varphi(u) < \infty$.

As a result of conditions (i)-(v), Geman and Reynolds recommend the family of functions, indexed by a parameter γ, and given by

$$\varphi_\gamma(u) = \frac{|u/\Delta|^\gamma}{1 + |u/\Delta|^\gamma}, \quad 0<\gamma\leq 1 \tag{3.4}$$

where Δ is a positive scaling parameter. This family can be seen to satisfy the conditions (i)-(v) ; Geman and Reynolds choose to work with the $\gamma=1$ member,

$$\varphi(u) = \frac{|u|/\Delta}{1 + |u|/\Delta}. \tag{3.5}$$

The smoothness penalty given by Equation (3.5) will be used for the remainder of this work. Figure 3.1 depicts the form of $\varphi(\,)$ for various values of Δ.

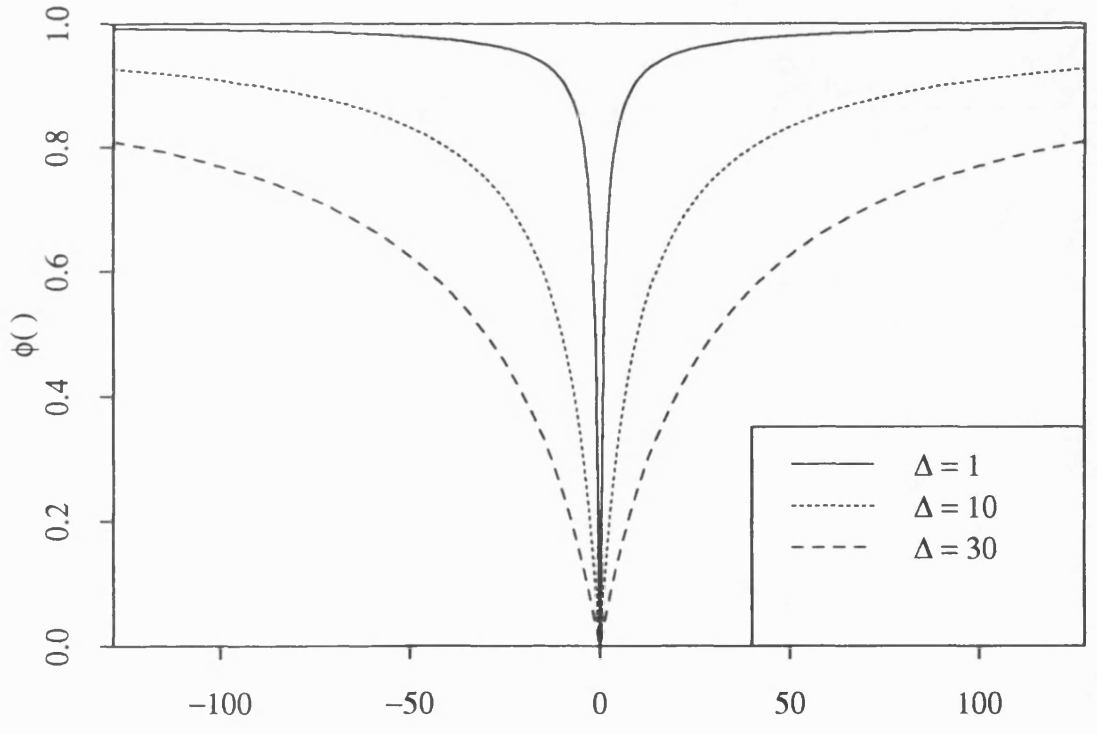


Figure 3.1 The $\varphi(u)$ in Equation (3.5) for various values of Δ.

There is an additional reason why using the $\varphi(\,)$ given by Equation (3.5) is beneficial for the recovery of discontinuities between smooth regions. One way in which this problem is sometimes tackled is by the introduction of a line process. Consider a simple case where we are interested in finding regions of constant grey level, using a smoothness measure of the form of Equation (3.2). A line process is an additional set of variables $\{\mathbf{L}_c\}$ defined on the clique interactions between

pixels. If there is a large absolute value of $D_c(\mathbf{X})$ for some clique c, suggesting that the two pixels involved belong to different regions, then we might wish to suspend our belief that these pixels are neighbours. We could separate these pixels by a line which breaks the clique; the line variables can take the values "on", 0, or "off", 1. Then, in calculating a modified penalty $\Phi^*(\mathbf{X},\mathbf{L})$, we would only count the contributions from pixels not separated by lines, $\Phi^*(\mathbf{X},\mathbf{L})=\sum_{c\in C}\mathbf{L}_c\varphi(D_c(\mathbf{X}))$. We hope that these contributions then reflect the degree of non-smooth behaviour due solely to degradation within regions. In this way, we are attempting to recover genuine discontinuities by not penalising them in the energy. Generally, an additional model would be used to incorporate interactions between lines, for example we might encourage connected lines, and penalise loose ends; a process including this organisational term is called an interacting line process. Further discussion of line processes may be found in Geman & Geman (1984) and Silverman, Jennison, Stander & Brown (1990). Geman and Reynolds prove that $\Phi(\mathbf{X})$, with the form of $\varphi(\ )$ given in Equation (3.5), is equal to the minimum, over $\{\mathbf{L}_c\}$, of a $\Phi^*(\mathbf{X},\mathbf{L})$ when $\varphi(u)=u^2$, and a particular non-interacting line process is incorporated. This implicitly involved line process is slightly different from the simple case we have described. The line variables are no longer dichotomous, they can take continuous non-negative values reflecting the strength of the associated clique. Also, $\Phi^*(\mathbf{X},\mathbf{L})$ does not include the additional model organising the lines. However, this $\Phi^*(\mathbf{X},\mathbf{L})$ does include a decreasing penalty $\psi(\mathbf{L}_c)$, imposed on $\mathbf{L}_c$, which prevents all the clique terms from being discounted in $\Phi^*(\mathbf{X},\mathbf{L})$. Such a model should have a beneficial effect on the recovery of discontinuities, and so the model using our choice of $\varphi(\ )$ will inherit these benefits.

3.3 Higher order models

In Equation (3.1), the total smoothness penalty was broken down into the sum of individual penalties for the smoothness of each clique. As an example of a measure of smoothness, Equation (3.2) gave a linear approximation to the first order derivative at adjacent pixels. We have chosen a penalty function $\varphi(D_c(\mathbf{X}))$, in Equation (3.5), which is increasing in $|D_c(\mathbf{X})|$ and satisfies $\varphi(0)=0$. So, a scene which has a low smoothness penalty using this smoothness measure, would consist of regions of constant grey-level, since within these regions, the approximation to the first order derivative will be zero. Unfortunately, it may be the case that the image to be restored cannot be adequately represented by such a model.

Geman & Reynolds (1992) consider extending the idea of smoothness to incorporate regions within which the pixel values can be related in some linear or quadratic fashion. To make this more precise, suppose we temporarily relabel the pixels (i,j) where i denotes the row, and j denotes the column of the pixel grid

(see Figure 2.1). Then suppose we have constants A, B, C, D, E and F. A region is constant if the pixel values $\mathbf{X}_{i,j}=A$ within the region. A region is planar if $\mathbf{X}_{i,j}=Ci+Bj+A$. A region is quadric if $\mathbf{X}_{i,j}=Fi^2+Ej^2+Dij+Ci+Bj+A$. If the pixel values are linearly related within some region, then a linear approximation to their second order derivative will be zero within this region. Similarly, if the pixel values are quadratically related, then a linear approximation to their third order derivative will be zero within the appropriate region. The obvious choice for a measure of smoothness is a linear approximation to the suitable level derivative. The models are in some sense nested; constant regions are an example of planar regions, which in turn, are an example of quadric regions. We will use the terminology first order to denote constant regions, and second and third order to denote planar and quadric regions respectively.

Geman and Reynolds describe the smoothness measures, and corresponding clique types, for models for planar and quadric images, derived from a first order model using the four nearest adjacent pixels as neighbours. We have extended the second and third order models, based on a first order model using the eight nearest adjacent pixels as neighbours. The first order cliques are formed in the obvious way, taking pixel differences at adjacent pixels. The second order cliques are formed by considering differences in first order $D_c(\ )$ at adjacent pixels. We have restricted the number of third order cliques by only considering the difference in second order collinear clique $D_c(\ )$ at adjacent pixels.

The clique types are illustrated on the following two pages (heavy solid lines are used to denote horizontal and vertical pixel adjacencies, dotted lines to denote diagonal adjacencies, and faint solid lines to "complete" the pixel grid). Each group of labelled pixels forms a clique. In the first order model, all cliques consist of two pixels. In the second order model, they consist of either three or four pixels, and in the third order, they consist of either four or six pixels. Obviously as the number of cliques increases, so does the computational complexity. In the first order model, eight cliques need to be considered for each pixel, in the second order model this increases to thirty-six, and in the third order eighty-eight. We will use the notation $D_c^i(\ )$ to indicate a linear approximation to the i^{th} order derivative. The expression for $D_c^2(\mathbf{X})$ in Equation (3.7) emphasizes the manner in which the cliques and smoothness measures have been formed.

$$D_c^1(\mathbf{X}) = \mathbf{X}_s - \mathbf{X}_t \qquad \text{Clique types (1), (2).} \tag{3.6}$$

$$D_c^2(\mathbf{X}) = \begin{cases} (\mathbf{X}_s-\mathbf{X}_t) - (\mathbf{X}_t-\mathbf{X}_u) & \text{Types (1), (3)} \\ (\mathbf{X}_s-\mathbf{X}_t) - (\mathbf{X}_u-\mathbf{X}_v) & \text{Types (2), (4), (5).} \end{cases} \tag{3.7}$$

$$D_c^3(\mathbf{X}) = \begin{cases} \mathbf{X}_s-3\mathbf{X}_t+3\mathbf{X}_u-\mathbf{X}_v & \text{Types (1), (3)} \\ \mathbf{X}_s-2\mathbf{X}_t+\mathbf{X}_u-\mathbf{X}_v+2\mathbf{X}_w-\mathbf{X}_x & \text{Types (2), (4), (5), (6).} \end{cases} \tag{3.8}$$

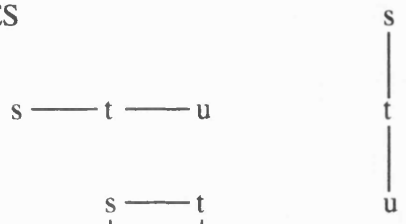
FIRST ORDER MODEL CLIQUES

Type (1)

Type (2)

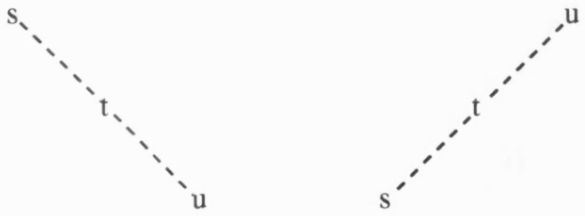
SECOND ORDER MODEL CLIQUES

Type (1)

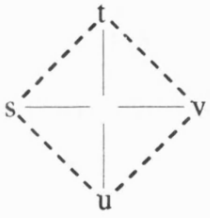


Type (2)

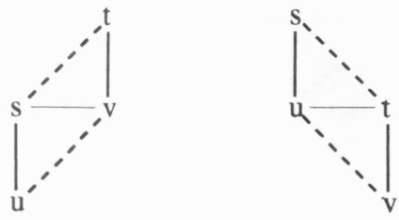
Type (3)



Type (4)

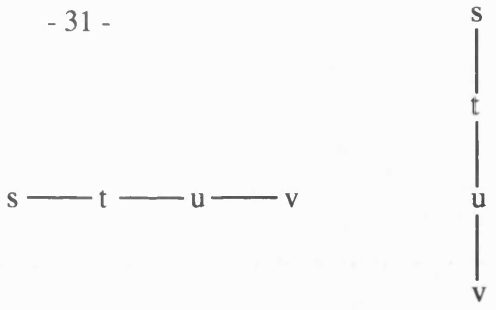


Type (5)

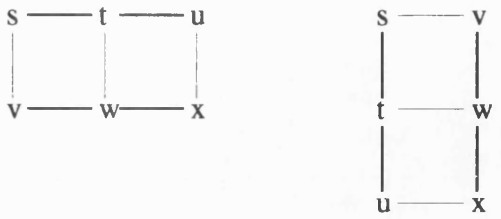


THIRD ORDER MODEL CLIQUES

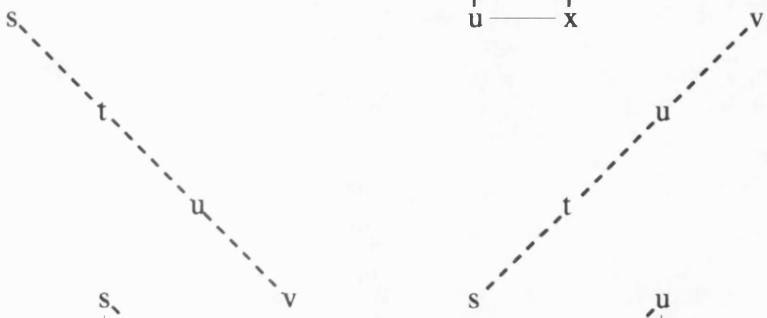
Type (1)



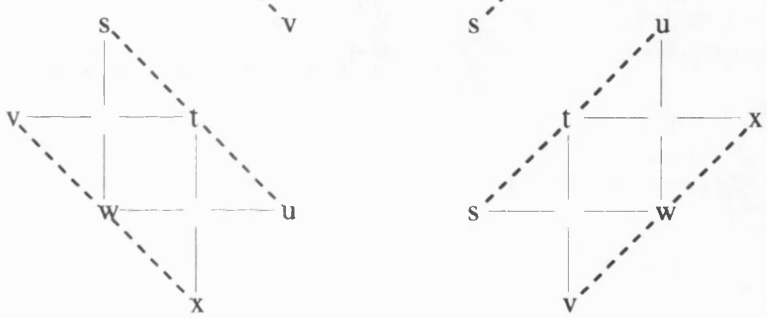
Type (2)



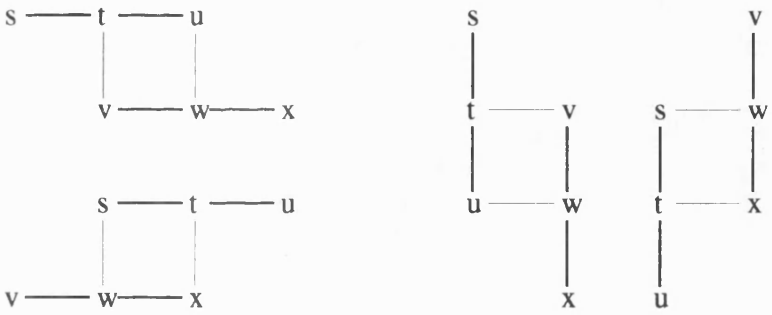
Type (3)



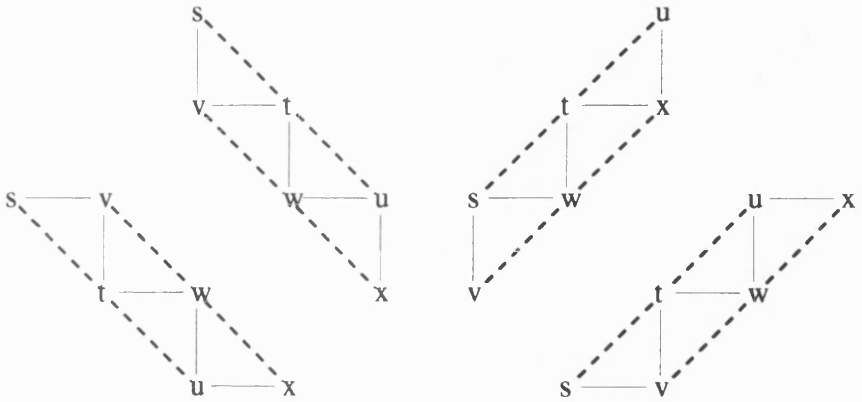
Type (4)



Type (5)



Type (6)



Geman and Reynolds' three models based on the four neighbour first order system consists of clique type (1) of the first order, types (1) and (2) of the second order, and types (1) and (2) of the third order, respectively. Their $D_c^i(\mathbf{X})$ are as defined for the appropriate cliques in Equations (3.6), (3.7) and (3.8).

Since the three orders of model have been extended to include diagonal pixel connections, the question of diagonal down-weighting arises. In the first order model, this is the situation where the contribution to the total smoothness penalty from a diagonal neighbour pair is given less weight than that from a horizontal or vertical neighbour pair. The use of down-weighting appears fairly common, reflecting the greater inter-pixel distance for the diagonal pairs in a square lattice; the distance between the centres of diagonally adjacent pixels is $\sqrt{2}$ times that between the centres of horizontally adjacent pixels. Jubb (1989) uses a down-weight of $1/\sqrt{2}$ applied to the contributions to the prior made by diagonal neighbours, and that practise is continued here. So, paired differences in the first order model have a weight of either 1 or $1/\sqrt{2}$ in $\Phi(\mathbf{X})$. The clique sums in the second order model correspond to differences between first order differences at adjacent pixels, hence the second order model introduces a second down-weight of either 1 or $1/\sqrt{2}$. We have chosen to apply this multiplicatively to the first order weighting. Similarly, the third order cliques correspond to differences between second order differences at adjacent pixels, and so on. Explicitly the weights which will be used are:

$$\text{First order weights : } \begin{cases} 1 & \text{Clique type (1)} \\ 1/\sqrt{2} & \text{Type (2).} \end{cases} \tag{3.9}$$

$$\text{Second order weights : } \begin{cases} 1 & \text{Types (1), (2)} \\ 1/\sqrt{2} & \text{Type (5)} \\ 1/2 & \text{Types (3), (4).} \end{cases} \tag{3.10}$$

$$\text{Third order weights : } \begin{cases} 1 & \text{Types (1), (2)} \\ 1/\sqrt{2} & \text{Type (5)} \\ 1/2 & \text{Type (6)} \\ 1/(2\sqrt{2}) & \text{Types (3), (4).} \end{cases} \tag{3.11}$$

The total smoothness constraint $\Phi(\mathbf{X})$ should now be written incorporating w_c^i, the appropriate weight for clique c of model order i. This has no effect in Geman and Reynolds' models, since the appropriate w_c^i are always 1.

$$\Phi(\mathbf{X}) = \sum_{c \in C} w_c^i \, \varphi(D_c^i(\mathbf{X})).$$

In Section 2.1, we described the formation of an energy function H(**X**). Incorporating the prior $\varphi(\,)$ given in Equation (3.5), and the three orders of model described in this section, we now have three energy functions,

$$H^i(\mathbf{X}) = \sum_{c \in C^i} w_c^i \frac{|D_c^i(\mathbf{X})| / \Delta^i}{1+|D_c^i(\mathbf{X})| / \Delta^i} + \lambda^i \|\mathbf{Y}-\mathbf{K}\mathbf{X}\|^2, \quad i=1, 2, 3, \quad (3.12)$$

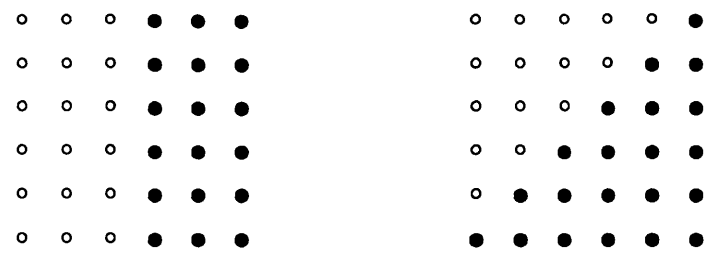
where i denotes the model order. Our stated aim in reconstruction is energy minimisation; there are now three different energy functions to minimise. Although these problems could be tackled independently, Geman and Reynolds generally advocate a sequential processing. The intention is that the restoration from a first order model will be passed as a starting value to the second order model, which in turn will produce a starting value for the third order model. Ideally the first order reconstruction will recover transitions between distinct first order regions, although it may also find extraneous transitions within higher order regions; visually, a smooth linear change in grey-level may be recovered as a series of terraces. Passing this reconstruction to a second order model, we are increasing the possible surfaces which the prior will regard as smooth, while allowing a better fit to the data. Regions which are genuinely first order, and the transitions between them, should be retained since a flat surface is a particular example of a planar surface. Similarly, third order improvements may be made to the second order reconstruction, while preserving genuine second and first order behaviour.

3.4 Parameter selection for Geman and Reynolds' prior

3.4.1 Rationale for selection

There are two parameters in each of the three energy functions, Equation (3.12), λ^i which balances the contributions from smoothness constraints and data fidelity, and Δ^i which scales the derivative approximations. Geman and Reynolds' intention is to select the parameters so that, for certain classes of simple image, the true image $\mathbf{X}^0$ is, at least, a coordinate-wise minimum of $H^i(\mathbf{X})$. A coordinate-wise minimum is defined to be an image satisfying the condition that a change at any one of its pixel values will result in an increase in its energy. If we can only update one pixel at a time, then all minima are coordinate-wise, but a coordinate-wise minimum may be local rather than global. A coordinate-wise descent algorithm such as single-site update ICM will converge to a such a minimum.

The class of $\mathbf{X}^0$ mainly considered by Geman and Reynolds are those consisting of a long horizontal or vertical step edge. This is a reasonable choice when working with the four neighbour first order model. Since we have extended the models to include diagonal interactions, we will also consider images consisting of a long diagonal step edge. The two types of edge are represented below:



The symbol • denotes a pixel taking the value 0, ∘ a pixel taking the value $J>0$. Notice that in $D_c^i(\mathbf{X})$, pixel values always appear as paired differences, so it is sufficient to consider these two scenes as representative of any magnitude J step edge of these types. Although these two images are first order, they will be used to determine parameter settings for all three model orders. We have stated in Section 3.3, that the restoration from the first order model may be passed as a starting value to the second order model, and so on. If these first order step edges are coordinate-wise minima of all three levels, then the higher order models should preserve first order discontinuities of this type.

Before we attempt to find parameter settings such that these step edges are coordinate-wise minima of all three energies, we will determine whether they are actually coordinate-wise minima of the prior alone. This is motivated by considering the ideal situation of an uncorrupted record, in which case $||\mathbf{Y}-\mathbf{KX}||^2=0$, and $H^i(\mathbf{X})=\Phi^i(\mathbf{X})$. Suppose, in this case, that these step edges were not coordinate-wise minima for any (Δ^i, λ^i). Then, since the prior is providing the impetus to deblur the scene (Section 3.2.2), it would seem unlikely that any suitable (Δ^i, λ^i) will exist when the data is blurred and noisy.

So, we are interested in a perturbation of size $u\neq 0$ at any single pixel s of the idealised image $\mathbf{X}^0$. This perturbation can be represented by an $|S^0|\times 1$ vector $\mathbf{u}^s$ taking the value zero at all coordinates except the s^{th}, where it takes the value u. Let $f_s^i(u)$ represent the change in the i^{th} order prior $\Phi^i(\,)$ due to the addition of $\mathbf{u}^s$ to $\mathbf{X}^0$. For $\mathbf{X}^0$ to be a coordinate-wise minimum of the prior $\Phi^i(\mathbf{X})$, we require that $f_s^i(u)>0$, $\forall\ s\in S^0$ and $u\neq 0$.

$$\begin{aligned} f_s^i(\,u\,) &= \Phi^i(\,\mathbf{X}^0+\mathbf{u}^s\,) - \Phi^i(\,\mathbf{X}^0\,) \\ &= \sum_{c:\ s\in c} w_c^i\,\{\varphi(\,D_c^i(\,\mathbf{X}^0+\mathbf{u}^s\,)) - \varphi(\,D_c^i(\,\mathbf{X}^0\,))\}. \end{aligned} \tag{3.13}$$

Geman and Reynolds find a set of strictly positive constants $\{c_i\}$ for the vertical step edge, under their model excluding diagonal interactions, such that $f_s^i(u)\geq c_i\varphi(u)$, $\forall u$, $s\in S^0$. Since $\varphi(u)>0$, $\forall\ u\neq 0$, this is sufficient to show that the vertical step edge is indeed a coordinate-wise minimum of their $\Phi^i(\mathbf{X})$.

Vertical step edge
no diagonal interactions

$$c_i = \begin{cases} 2, & i=1 \\ 5, & i=2 \\ 14, & i=3. \end{cases}$$

Similarly, we can find strictly positive constants $\{c_i\}$ for the diagonal step edge, under our extended model including diagonal interactions, such that $f_s^i(u) \geq c_i \varphi(u)$, $\forall u$, $s \in S^0$.

$$\text{Diagonal step edge, diagonal interactions} \quad c_i = \begin{cases} \sqrt{2}, & i=1 \\ 8/\sqrt{2} - 3, & i=2 \\ 44/(3\sqrt{2}) - 13/4, & i=3 \quad \text{provided } J/\Delta \geq 1 \ . \end{cases} \tag{3.14}$$

The calculations to find these constants are given in Appendix 3.5. The additional constraint necessary for the third order model indicates the diagonal step edge's lower stability. For the $i=1$ and $i=2$ models, c_i is independent of J suggesting that an arbitrarily small edge can be a stable minimum of the model. For $i=3$, there is a size of step edge below which certain perturbations may produce an alternative image more acceptable to the prior. However, provided that this condition is met for the third order model, the diagonal step edge is also a coordinate-wise minimum of this $\Phi^i(\mathbf{X})$.

3.4.2 Choice of smoothing parameter λ

The aim in selecting λ is to ensure that the simple step edges $\mathbf{X}^0$, described in the last section, are coordinate-wise minima of the energy function $H^i(\ )$, Equation (3.12). In fact, the set of coordinate-wise minima will be a random set because the energy function involves the random noise; the best that we can manage in this situation, is to ensure that the probability of $\mathbf{X}^0$ being in this random set is arbitrarily close to 1. The calculations involved in achieving this aim are quite involved, and as a result this section is fairly lengthy; for this reason, we will now provide a brief outline of the whole section.

The conditions for $\mathbf{X}^0$ to be a coordinate-wise minimum of the energy are stated, and the event that these constraints are met is defined. The dimension of the problem requires us to work with a set of constituent events, these events being that the constraints are met pixel-wise. The weaker conditions for these constituent events to occur can be rearranged in terms of an inequality in the components of the noise realisation. Since the distribution of the noise is known, we can express a minimum probability for each of these events to occur. In order to guarantee that the main event occurs with a certain probability $1-\varepsilon$, we then apply the Bonferroni inequality after guaranteeing that each of the constituent events occurs with a related, greater probability. Linking the two resulting probability inequalities, we are left with an upper bound for λ^i; any positive λ^i less than this bound will guarantee that the $\mathbf{X}^0$ will be a coordinate-wise minimum of the i^{th} order energy with a probability of at least $1-\varepsilon$.

As a first step, we are interested in considering the changes in the energy due to some perturbation $\mathbf{u}^s$, the perturbation of any pixel $s \in S^0$ by amount u. For notational simplicity, the model order i will be dropped whenever an expression applies equally to all three orders. In order to emphasize the random nature of the realisations, the Normal noise $\boldsymbol{\eta}$ is reintroduced. Then, rewriting the energy function explicitly stressing $\boldsymbol{\eta}$, using the weights in Equations (3.9)-(3.11), and Equation (2.2), $\mathbf{Y}=\mathbf{K}\mathbf{X}^0+\boldsymbol{\eta}$,

$$H(\mathbf{X},\boldsymbol{\eta}) = \sum_{c \in C} w_c \varphi(D_c(\mathbf{X})) + \lambda \| \mathbf{K}\mathbf{X}^0+\boldsymbol{\eta}-\mathbf{K}\mathbf{X} \|^2. \tag{3.15}$$

We will denote the change in the energy function due to the perturbation $\mathbf{u}^s$ by

$$\delta_{s,u}H(\boldsymbol{\eta}) = H(\mathbf{X}^0+\mathbf{u}^s,\boldsymbol{\eta}) - H(\mathbf{X}^0,\boldsymbol{\eta}). \tag{3.16}$$

The event of interest, to be denoted Λ, is that the noise realisation is such that $\mathbf{X}^0$ is a coordinate-wise minimum of $H(\mathbf{X},\boldsymbol{\eta})$. In order for this to be the case, $\delta_{s,u}H(\boldsymbol{\eta})$ must be positive for any non-zero perturbation u at any pixel $s \in S^0$, so

$$\Lambda = \{ \boldsymbol{\eta} \mid \delta_{s,u}H(\boldsymbol{\eta})>0, \forall s \in S^0, \forall u \neq 0 \}. \tag{3.17}$$

The aim is then to find conditions on λ in terms of Δ, the blurring $\mathbf{K}$, the true scene $\mathbf{X}^0$, and the noise variance σ^2, such that $P_\sigma(\Lambda) \approx 1$.

The approach taken by Geman and Reynolds in tackling this high dimensional problem is to divide the event Λ into the set of smaller pixel-wise events Λ_s. These events are that the noise realisation is such that, at a particular pixel s, the non-zero perturbation u will result in an increase in energy,

$$\Lambda_s = \{ \boldsymbol{\eta} \mid \delta_{s,u}H(\boldsymbol{\eta})>0, \forall u \neq 0 \}, \quad s \in S^0. \tag{3.18}$$

Since $\Lambda = \bigcap_{s \in S^0} \Lambda_s$, if we can determine the probability of each Λ_s, then we can apply the Bonferroni inequality to obtain a lower bound on the probability of Λ occurring.

We need to consider the change in energy due to a perturbation $\mathbf{u}^s$, for a particular pixel $s \in S^0$. This change is denoted $\delta_{s,u}H(\boldsymbol{\eta})$ in Equation (3.16) and uses the definition of the energy given in Equation (3.15). We are looking to find the probability that this change is positive for non-zero u. Before expanding the expression, it might be helpful to recall two items of notation used in earlier sections. The difference in the prior $\Phi(\,)$ created by the perturbation $\mathbf{u}^s$ was denoted by $f_s(u)$ in Equation (3.13) (the model order i has been dropped). Also, the action of the blurring matrix $\mathbf{K}$ on the vector $\mathbf{X}$, for pixel s, can be written as the sum over pixels involved in the blurring of s (the set B_s) of pixel values multiplied by the relevant blurring coefficients, $(\mathbf{K}\mathbf{X})_s = \sum_{t \in B_s} \gamma_{s-t}\mathbf{X}_t$ (Equation (2.1)).

$$\delta_{s,u}H(\eta) = \sum_{c\in C} w_c\{\varphi(D_c(\mathbf{X}^0+\mathbf{u}^s))-\varphi(D_c(\mathbf{X}^0))\}+\lambda\{\|\mathbf{KX}^0+\eta-\mathbf{K}(\mathbf{X}^0+\mathbf{u}^s)\|^2-\|\eta\|^2\}$$

$$= f_s(u) + \lambda \{ \| \eta-\mathbf{K}\mathbf{u}^s\|^2 - \| \eta \|^2\}$$

$$= f_s(u) + \lambda \sum_{t\in S} \{ (\eta_t - \sum_{r\in B_t} \gamma_{t-r}\mathbf{u}^s_r)^2 - \eta_t^2\}$$

$$= f_s(u) + \lambda \sum_{t\in S} \{ \gamma^2_{t-s}u^2 - 2\eta_t\gamma_{t-s}u \}$$

$$= f_s(u) + \lambda (\beta_s u^2 - 2uZ_s(\eta)) \qquad (3.19)$$

where $\beta_s=\sum_{t\in S}\gamma^2_{t-s}$, and $Z_s(\eta)=\sum_{t\in S}\gamma_{t-s}\eta_t$. Since γ_{t-s} corresponds to the weight given to the value of pixel s in the blurring of pixel t, β_s equals the sum of squared weights for pixel s contributions to the blurred values $(\mathbf{KX})_t$, for which $s\in B_t$. All pixels contribute to some blurred value, even if they themselves lie in $S^0\backslash S$ and have no record, therefore $\beta_s>0$, $\forall s\in S^0$. The maximum value attainable by β_s will occur when all the blurred values to which s could contribute lie in S; we will denote this maximum by β.

The event of interest is that $\delta_{s,u}H(\eta)>0$ for all non-zero u. Using Equation (3.19), this is equivalent to the event that,

$$0 < f_s(u) + \lambda (\beta_s u^2 - 2uZ_s(\eta)) , \quad u\neq 0$$

$$\Rightarrow \qquad u\,Z_s(\eta) < (2\lambda)^{-1}(f_s(u) + \lambda\beta_s u^2) , \qquad u\neq 0.$$

We have a lower bound on $f_s(u)$ from Section 3.4.1, namely $f^i_s(u)\geq c_i\varphi(u)$; the $\{c_i\}$, i=1, 2, 3, are given in Equation (3.14) for a diagonal step edge under our extended model including diagonal interactions. So the event Λ_s certainly occurs if

$$uZ_s(\eta) < \frac{c_i\varphi(u) + \lambda\beta_s u^2}{2\lambda}.$$

The right hand side of the inequality is even in u, so we require $Z_s(\eta)$ to lie between $\pm(c_i\varphi(u)+\lambda\beta_s u^2)/(2u\lambda)$, evaluated for $u>0$. We can see from the definition of $Z_s(\eta)$, as a linear combination of the $N(0,\sigma^2)$ noise components, that it is distributed as a $N(0,\beta_s\sigma^2)$ variate. We are now at the stage where we have an expression involving a variable of known distribution which implies the pixel-wise event in which we are interested.

$$|Z_s(\eta)| < \inf_{u>0} \left\{ \frac{c_i\varphi(u) + \lambda\beta_s u^2}{2\lambda u} \right\} \Rightarrow \delta_{s,u}H^i(\eta) > 0, \ \forall u\neq 0.$$

In order to evaluate the minimum in the $|Z_s(\eta)|$ expression above, let $F_s^i(u)=(2\lambda u)^{-1}(\lambda\beta_s u^2+c_i\varphi(u))$. Then we are interested in finding $L_s^i=\inf_{u>0} F_s^i(u)$. To do this, we will use the specific form of $\varphi(\,)$ given in Equation (3.5), so

$$F_s^i(u) = \frac{1}{2\lambda u}\left\{\lambda\beta_s u^2 + \frac{c_i u/\Delta}{1+u/\Delta}\right\}$$

$$= \frac{\beta_s u}{2} + \frac{c_i}{2\lambda(1+u/\Delta)\Delta}.$$

Differentiating $F_s^i(u)$ with respect to u gives a function increasing in positive u,

$$F_s^{i\prime}(u) = \frac{\beta_s}{2} - \frac{c_i}{2\lambda(1+u/\Delta)^2\Delta^2}.$$

If there is a local minimum of $F_s^i(u)$, then the minimising u will satisfy,

$$F_s^{i\prime}(u) = 0$$

$$\Rightarrow \qquad (u+\Delta)^2 = \frac{c_i}{\beta_s\lambda}$$

$$\Rightarrow \qquad u = -\Delta \pm \left[\frac{c_i}{\beta_s\lambda}\right]^{1/2}.$$

We have specified that we are only considering positive u. So, this pair of equations will give rise to one valid solution for u provided that $-\Delta+\sqrt{c_i/(\beta_s\lambda)}>0$, that is $\Delta<\sqrt{c_i/(\beta_s\lambda)}$. The corresponding minimum value of $F_s^i(\,)$ is then $(\beta_s c_i/\lambda)^{1/2}-\beta_s\Delta/2$. If $\Delta>\sqrt{c_i/(\beta_s\lambda)}$, then there is no local minimum for positive u; the required L_s^i will occur at the endpoint $u=0$, since in this case $F_s^{i\prime}(u)$ is strictly positive for $u>0$, implying that $F_s^i(u)$ is increasing. In this situation, the corresponding minimum value of $F_s^i(\,)$ is $c_i/(2\lambda\Delta)$. Rearranging the condition on Δ in terms of λ, these two results can be expressed,

$$L_s^i = \begin{cases} \dfrac{c_i}{2\lambda\Delta}, & \lambda \geq \dfrac{c_i}{\beta_s\Delta^2} \\ \left[\dfrac{\beta_s c_i}{\lambda}\right]^{1/2} - \dfrac{\beta_s\Delta}{2}, & \lambda < \dfrac{c_i}{\beta_s\Delta^2}. \end{cases} \tag{3.20}$$

The probability of the event Λ_s has already been shown to be greater than the probability of the event that $|Z_s(\eta)|<L_s^i$. Equation (3.20) gives an expression for L_s^i, and we know the distribution of $Z_s(\eta)$ to be $N(0,\beta_s\sigma^2)$, so

$$P_\sigma(\Lambda_s) > P_\sigma(|Z_s(\eta)| < L_s^i)$$

$$= \frac{2}{\sqrt{2\pi}}\int_0^{L_s^i/\sqrt{\beta_s\sigma^2}} e^{-t^2/2}\,dt. \tag{3.21}$$

The main event of interest is Λ, the event that the conditions for $\mathbf{X}^0$ to be a coordinate-wise minimum of the energy are not violated at any pixel. This event is the intersection over $s \in S^0$ of all the events Λ_s. The lower bound on the probability of Λ_s occurring, given by Equation (3.21), depends on the pixel s only through β_s, which appears in the upper limit of integration explicitly, and also implicitly through L_s^i. We could eliminate this dependence on s by finding the minimum, over s, of this upper limit of integration, $L_s^i/\sqrt{\beta_s\sigma^2}$. This would give a lower bound on $P_\sigma(\Lambda_s)$ independent of the particular pixel s. Consider the two ranges (i) $\beta_s \geq c_i/(\lambda\Delta^2)$ and (ii) $\beta_s < c_i/(\lambda\Delta^2)$, which correspond to the two intervals for the two formulae for L_s^i in Equation (3.20). Then it can trivially be seen that $L_s^i/\sqrt{\beta_s\sigma^2}$ is minimised by maximising β_s. The maximum value of β_s has already been denoted by β; it occurs when all the blurred values to which s could contribute lie in S. We will use the notation L^i to denote the L_s^i calculated according to Equation (3.20) with this value β. Then a pixel-independent lower bound on the probability of event Λ_s occurring is

$$P_\sigma(\Lambda_s) > \frac{2}{\sqrt{2\pi}} \int_0^{L^i/\sqrt{\beta\sigma^2}} e^{-t^2/2}\, dt, \quad \forall\, s \in S^0. \tag{3.22}$$

The analysis given by Geman and Reynolds assumes throughout that the pixel s is such that $\beta_s = \beta$; our treatment retains β_s for all $s \in S^0$ up to this point, and then justifies removing the dependence on s.

We now have a lower bound on the Λ_s probabilities, and although this is interesting in itself, the intention is to find a bound on the probability of the event Λ occurring. Since $\Lambda = \bigcap_{s \in S^0} \Lambda_s$, the Bonferroni inequality can be applied to the probabilities of the Λ_s, to give a lower bound on the probability of Λ. This inequality implies that, to guarantee the event Λ with probability $1-\varepsilon$ for some $\varepsilon > 0$, the individual events Λ_s must be guaranteed with probability $1 - \frac{\varepsilon}{|S^0|}$, where $|S^0|$ is the total number of pixels.

$$P_\sigma(\Lambda_s) > 1 - \frac{\varepsilon}{|S^0|}, \ \forall s \in S^0 \quad \Rightarrow \quad P_\sigma(\Lambda) > 1 - \varepsilon.$$

We have already shown that the event Λ_s occurs with a minimum probability given in Equation (3.22). We require that this minimum probability is at least $1-\varepsilon/|S^0|$. Suppose we reparameterise, replacing ε by $d = d(\varepsilon)$ defined by

$$\frac{2}{\sqrt{2\pi}} \int_0^d e^{-t^2/2}\, dt = 1 - \frac{\varepsilon}{|S^0|}. \tag{3.23}$$

Then, by considering the ranges of integration in Equations (3.22) and (3.23), with L^i equal to the L_s^i defined in Equation (3.20) calculated with $\beta_s = \beta$, we know that

d must satisfy the condition that $d \leq \frac{L^i}{\sqrt{\beta\sigma^2}}$. Since there are two formulae for L^i, depending on the value of λ, two cases must be considered.

(i) $$L^i = \frac{c_i}{2\lambda\Delta} \qquad \text{when } \lambda \geq \frac{c_i}{\beta\Delta^2}.$$

Then we require that $$d \leq \frac{c_i}{2\lambda\Delta\sqrt{\beta}\sigma}, \qquad \lambda \geq \frac{c_i}{\beta\Delta^2}$$

$$\Rightarrow \qquad \lambda \leq \frac{c_i}{2\Delta\sqrt{\beta}\sigma d}, \qquad \lambda \geq \frac{c_i}{\beta\Delta^2}.$$

For some valid λ to exist, satisfying both of the above inequalities on its range,

$$\frac{c_i}{\beta\Delta^2} \leq \frac{c_i}{2\Delta\sqrt{\beta}\sigma d}$$

$$\Leftrightarrow \qquad \sigma \leq \frac{\Delta\sqrt{\beta}}{2d}.$$

So, if the standard deviation of the noise variance satisfies $\sigma \leq (\Delta\sqrt{\beta})/(2d)$, then selecting $c_i/(\beta\Delta^2) \leq \lambda \leq c_i/(2\Delta\sqrt{\beta}\sigma d)$ will ensure that the step edges $\mathbf{X}^0$ are coordinate-wise minima of the energy with probability of at least $1-\varepsilon$.

(ii) $$L^i = \left[\frac{\beta c_i}{\lambda}\right]^{1/2} - \frac{\beta\Delta}{2} \qquad \text{when } \lambda < \frac{c_i}{\beta\Delta^2}.$$

Then we require that $$d \leq \left[\frac{c_i}{\lambda\sigma^2}\right]^{1/2} - \frac{\sqrt{\beta}\Delta}{2\sigma}, \qquad \lambda < \frac{c_i}{\beta\Delta^2}$$

$$\Rightarrow \qquad \lambda \leq \frac{c_i}{(d\sigma + \Delta\sqrt{\beta}/2)^2}, \qquad \lambda < \frac{c_i}{\beta\Delta^2}.$$

So, we now have two upper bounds on λ. To find the conditions under which $c_i/(\beta\Delta^2)$ is the larger of the two bounds,

$$\frac{c_i}{\beta\Delta^2} > \frac{c_i}{(d\sigma + \Delta\sqrt{\beta}/2)^2}$$

$$\Leftrightarrow \qquad \sigma > \frac{\Delta\sqrt{\beta}}{2d}.$$

So, if the standard deviation of the noise variance satisfies $\sigma > (\Delta\sqrt{\beta})/(2d)$, then selecting any $\lambda \leq c_i/(d\sigma + \Delta\sqrt{\beta}/2)^2$ will ensure that the step edges $\mathbf{X}^0$ are coordinate-wise minima of the energy with probability of at least $1-\varepsilon$. When $\sigma \leq (\Delta\sqrt{\beta})/(2d)$, then the smaller of the two limits on λ is $c_i/(\beta\Delta^2)$. Combining this with situation (i), then when the noise variance satisfies this condition, any $\lambda \leq c_i/(2\Delta\sqrt{\beta}\sigma d)$ will ensure that the step edges $\mathbf{X}^0$ are coordinate-wise minima of the energy with probability of at least $1-\varepsilon$.

In summary, we have found an upper bound on the i^{th} order model λ depending on the known blurring and noise variance, and also on the parameters Δ and d. If λ^i is chosen below the appropriate upper limit, then the step edges $\mathbf{X}^0$ will be coordinate-wise minima of the three energy functions with at least the probability determined by d, via Equation (3.23). These upper limits are given below,

$$\lambda^i \leq \begin{cases} \dfrac{c_i}{2\Delta\sqrt{\beta}\sigma d}, & \sigma \leq \dfrac{\Delta\sqrt{\beta}}{2d} \\ \dfrac{c_i}{(d\sigma+\Delta\sqrt{\beta}/2)^2}, & \sigma > \dfrac{\Delta\sqrt{\beta}}{2d}. \end{cases} \tag{3.24}$$

The problem of selecting the two parameters (λ,Δ) has now been reformulated in terms of selecting the two parameters (Δ,d). We will discuss the selection of Δ in the next section. The parameter d is defined in Equation (3.23) as the $1-\varepsilon/(2|S^0|)$ quantile of a standard Normal. Here $1-\varepsilon$ is the minimum probability that the step edge $\mathbf{X}^0$ is a coordinate-wise minimum of the energy. Geman and Reynolds point out that the analysis, in particular finding the constants $\{c_i\}$, is based on the worst case scenario; they suggest replacing $|S^0|$ in Equation (3.23) by a substantially smaller figure, and recommend the value d=3. More will be said about the selection of d in Section 3.4.5.

3.4.3 Choice of scaling parameter Δ

The parameter Δ is interpreted as a scaling parameter in the prior terms of the energy function. Working with the $\varphi(\ )$ given in Equation (3.5),

$$\varphi(\ D_c^i(\mathbf{X})\ ) = \frac{|D_c^i(\mathbf{X})|\ /\ \Delta}{1 + |D_c^i(\mathbf{X})|\ /\ \Delta}$$

$$= 1 - \frac{1}{1 + |D_c^i(\mathbf{X})|\ /\ \Delta}.$$

The effect on $\varphi(\ )$ of different values of Δ was seen in Figure 3.1. If Δ is small, then the penalty for non-zero derivative approximations will rise sharply but also level off quickly. In reconstruction, this may result in overpenalising variability within regions, although it should recover genuine discontinuities well. Conversely, when Δ is large, the penalty will only be gradually increasing and will begin to level off for much larger values of $|D_c^i(\mathbf{X})|$. This might interfere with both the deblurring, and the discontinuity recovery. The appropriate value of Δ will depend on the size of discontinuity considered important. In turn, this depends upon the number of grey-levels; a step of 4 grey-levels in a 64 level scene may be as important as a step of 16 grey-levels in a 256 level scene.

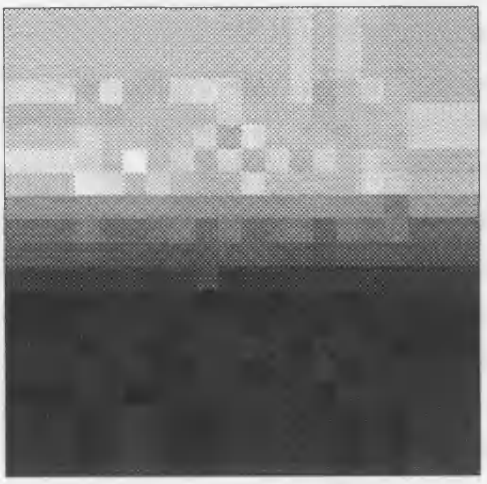
Geman and Reynolds choose to vary Δ between the three models. They say that the "choice of Δ is ad hoc. In a standard image with 256 grey levels, it seems reasonable that an edge of 20 to 30 grey levels is significant. On the other hand, a change of 2 or 3 in the *slope* of a planar surface is visually significant." They then recommend the values $\Delta \approx 10-20$ in the first order model and $\Delta \approx 3-10$ in the second and third order models. Experimental results support this reasoning. Notice that in the third order model under the diagonal scheme, we have the restriction that $J/\Delta \geq 1$, where J is the size of the discontinuity in $\mathbf{X}^0$. Lowering Δ between the levels ensures that $J/\Delta \geq 1$ for all but the smallest of step edges. As with the parameter d, further comments will be made about the selection of Δ in Section 3.4.5, after considering some reconstructions.

3.4.4 Examples

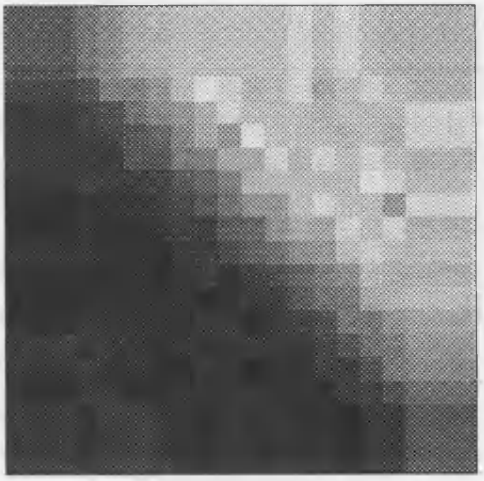
The intention of the examples in this section is to demonstrate Geman and Reynolds' prior and parameter selection in conjunction with a variety of images. The first two test scenes are simple step edges of the types upon which the parameter selection is based. The remaining three scenes are more testing examples of a first, second and higher order image respectively, as discussed in Section 3.3.

We will first consider the two step edge images, one horizontal and one diagonal. The original scenes are 20×20 pixels in dimension, and have a range of 64 grey-levels; the step edges are of size 20. Both scenes are corrupted by 5×5 uniform blurring and $N(0,1)$ noise; their records are shown in Figure 3.2(a) and (b). There are two points which should be made about the method of display. The first is that the grey-scale has been truncated so that the overall lowest and highest occurring pixel values of the six images are displayed as white and black respectively; in this case these extremes are the values 18 and 45. This accentuates features of the reconstructions, and is useful in detecting patterns of misclassification. The second point is that the record is displayed as a 20×20 scene although there will only be 16×16 records after blurring. Here values are extrapolated out to the image edges by redisplaying the value of the closest existing record in the grid. This convention is also used to generate a starting configuration for the restoration stage. For notational simplicity, we will extend the terminology four or eight neighbour model from first order models to the higher order models, indicating whether or not diagonal adjacencies are excluded.

Two first order reconstructions are shown for each of the images. The first uses the four neighbour model given by Geman and Reynolds, the second our extended eight neighbour model. The parameter λ has been selected using the approach described in Section 3.4.2, using the $\{c_i\}$ appropriate to each model. Equation (3.24) gives an upper bound on λ, any strictly positive value below this

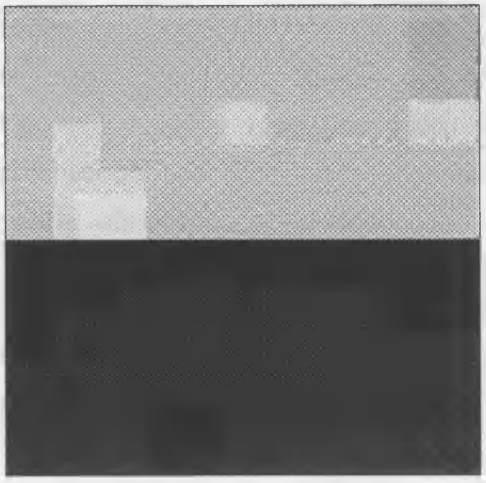


(a) Record for horizontal step edge

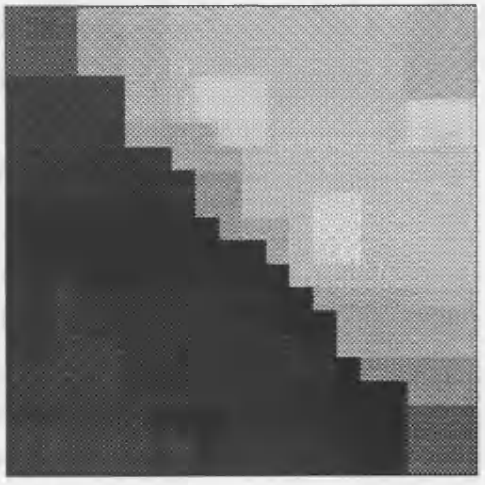


(b) Record for diagonal step edge

Normal $N(0,1)$ noise and 5×5 uniform blurring

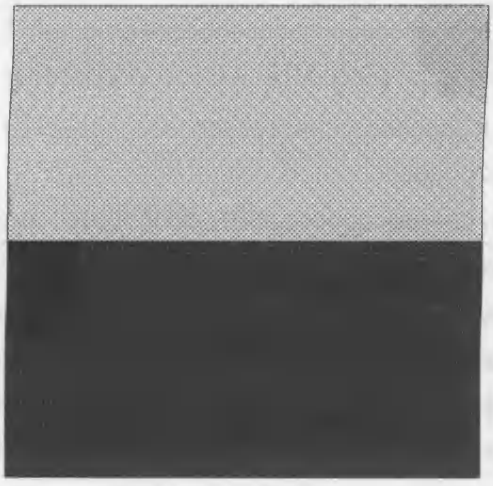


(c) Reconstruction of horizontal step edge

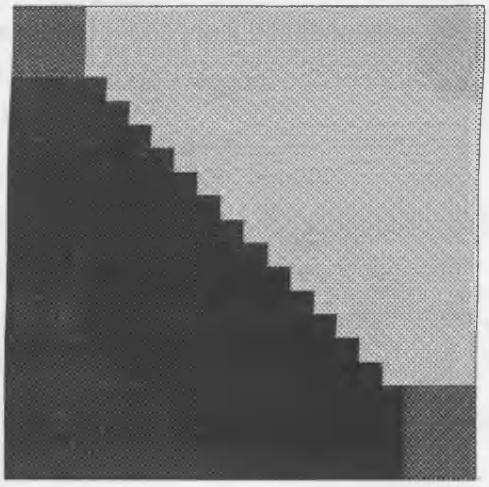


(d) Reconstruction of diagonal step edge

Four neighbour model, d=3 and Δ=15



(e) Reconstruction of horizontal step edge



(f) Reconstruction of diagonal step edge

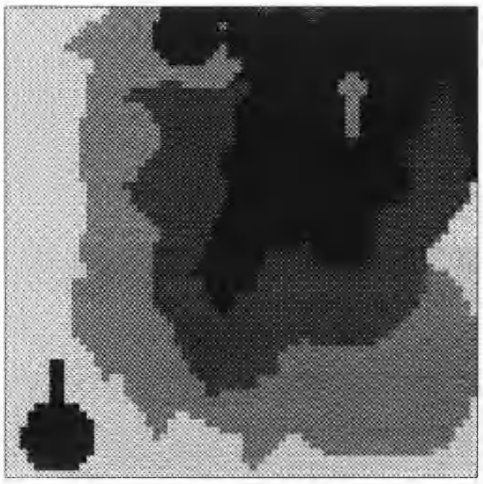
Eight neighbour model, d=3 and Δ=15

Figure 3.2 First order ICM reconstructions of two test scenes (scaled)

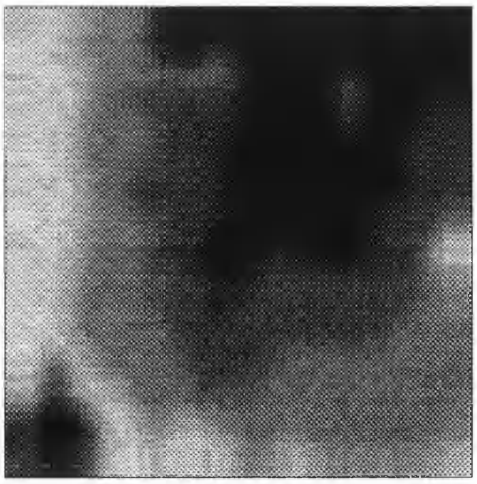
bound is acceptable; we have chosen to work with the limiting case (replacing the inequality in Equation (3.24) by an equality). The parameters Δ and d have been set to the values 15 and 3 respectively. The reconstruction technique used is single-site ICM; this is guaranteed to converge to some, not necessarily good, coordinate-wise minimum. Comparing Figures 3.2(c) and (d) with (e) and (f), it is clear that the eight neighbour model has been more successful in reconstructing both types of step edges. The four neighbour model mainly recovers the horizontal step edge, but tends to force the diagonal edge into fewer, steeper sections of slope. The eight neighbour model recovers the main part of both edges. Both models suffer problems with the pixels in $S^0 \backslash S$. Intuitively, the eight neighbour model should be able to deal with a horizontal or vertical edge, since the diagonal adjacency information may increase such an edge's stability. This can be confirmed analytically, by showing that the c_1 corresponding to a vertical step edge under an eight neighbour model is 2. Since the eight neighbour c_1 for a diagonal edge is $\sqrt{2}<2$, and in Equation (3.24) the upper bound $\lambda \propto c_1$, the diagonal edge based λ satisfies the equivalent inequality for a vertical edge. No energy comparison has been made between reconstructions since the two models work with a different energy function.

The three more complex test images are shown in Figure 3.3(a), (c) and (e). They have been chosen to be non-trivial examples of a first, second and, at least, third order scene. These images are all 64×64 pixels in dimension. The first order scene has 64 grey-levels, the second and higher order scenes have 256 grey-levels. The intention of these examples is to demonstrate the different model orders, and the hierarchy of processing discussed in Section 3.3. All three model orders have been used on each image, and the reconstruction at each level is passed as the starting value to the next level. In line with the examples in Geman and Reynolds, the records generated from these test scenes are heavily degraded with blurring, but with relatively low noise levels. These records are shown in Figure 3.3(b), (d) and (f), which also details the degradation.

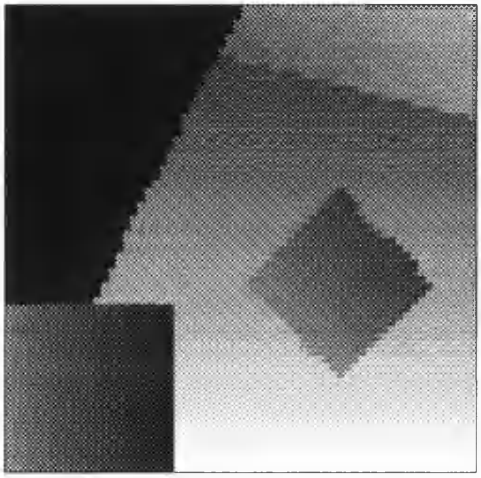
Consider the six reconstructions of the first order scene shown in Figure 3.4. Both first order models largely succeed in deblurring the scene, although the eight neighbour model appears to be more successful than the four neighbour model at identifying the region boundaries. The eight neighbour model also appears to be producing smoother reconstructions within regions. It can be seen that applying the second and third order models to the first order reconstructions has little effect. One possible exception is where an edge has not been recovered successfully by the first order model. In this case, the two higher order models possibly increase the error by attempting to smooth over the discontinuity, either linearly or quadratically. One unexpected problem with the reconstructions is the



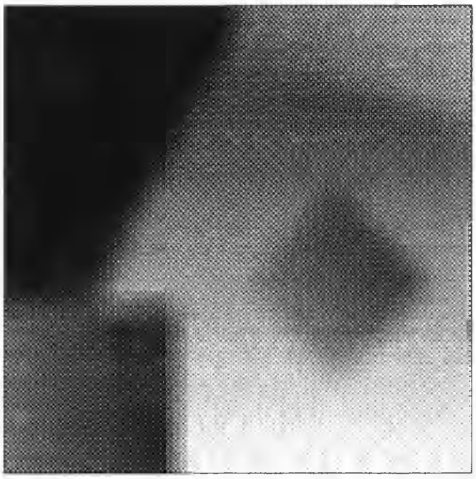
(a) A first order test scene
64 grey-levels



(b) Record after 7×7 Gaussian
blur and $N(0,1)$ noise

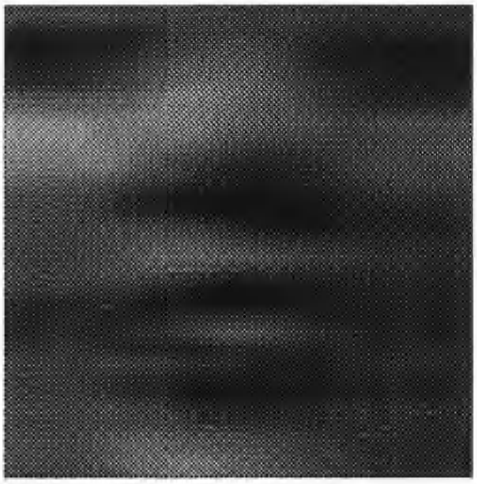


(c) A second order test scene
256 grey-levels



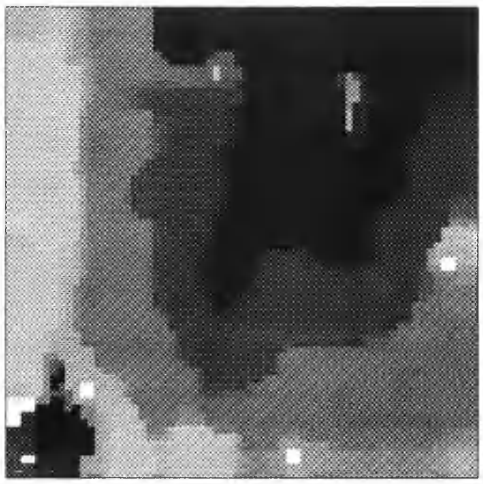
(d) Record after 5×5 uniform
blur and $N(0,4)$ noise

(e) A higher order test scene
256 grey-levels

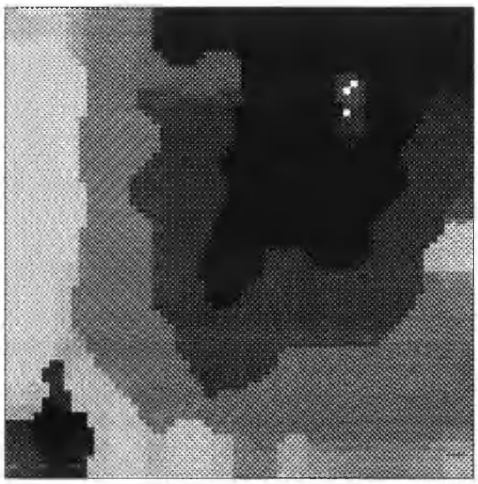


(f) Record after 1×21 motion
blur and $N(0,0.25)$ noise

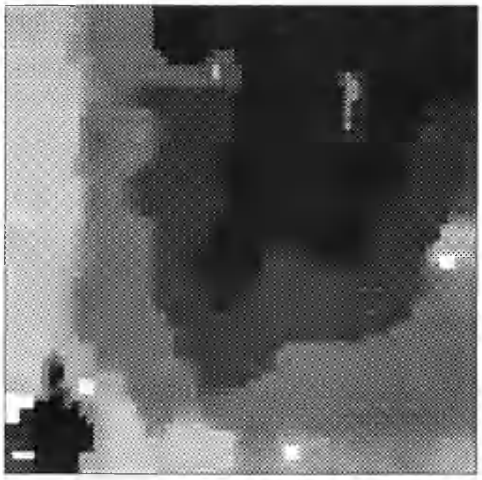
Figure 3.3 The three test scenes and their records



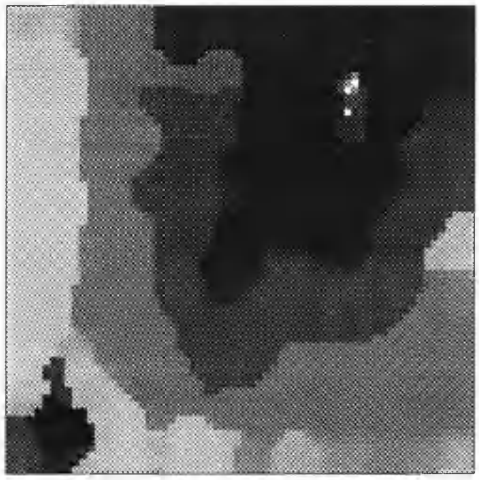
(a) First order, 4 neighbour, Δ=15 d=3



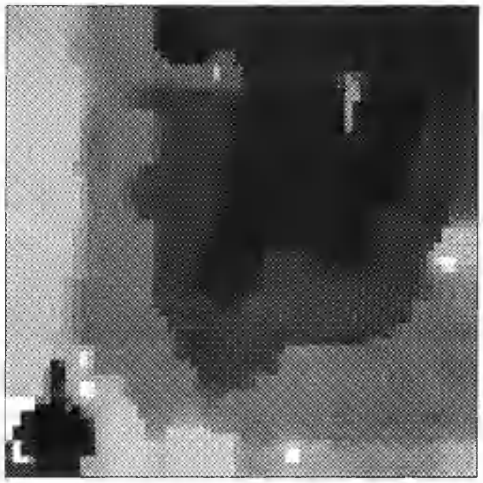
(b) First order, 8 neighbour, Δ=15 d=3



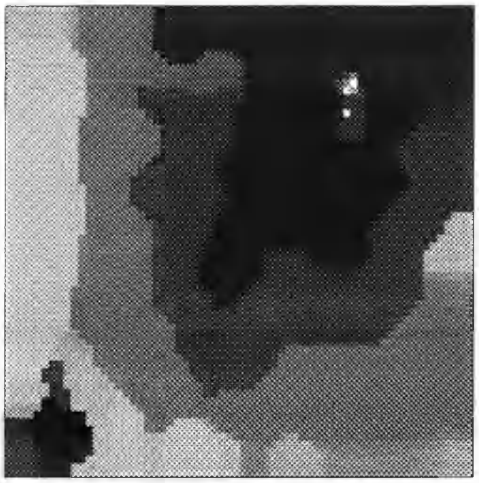
(c) Second order, 4 neighbour, Δ=2 d=3



(d) Second order, 8 neighbour, Δ=2 d=3

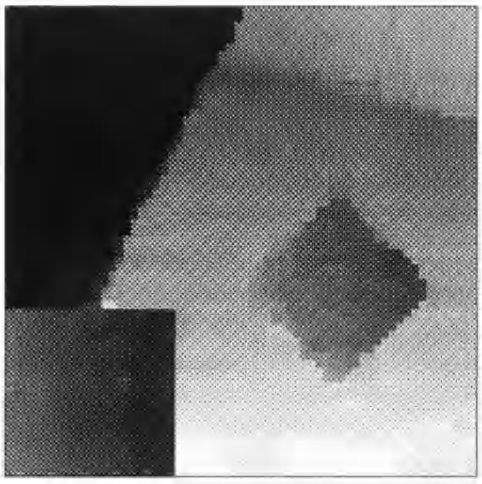


(e) Third order, 4 neighbour, Δ=2 d=3

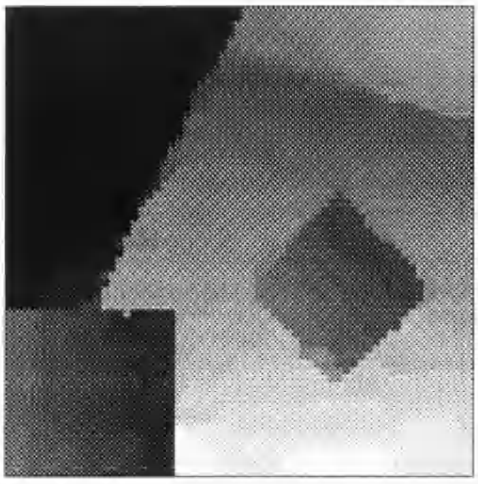


(f) Third order, 8 neighbour, Δ=2 d=3

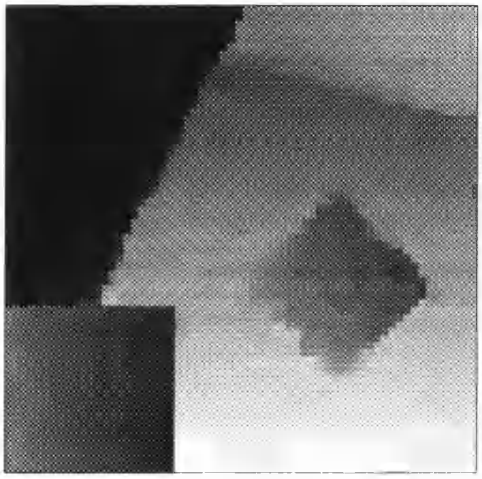
Figure 3.4 Three levels of ICM reconstruction of a first order scene



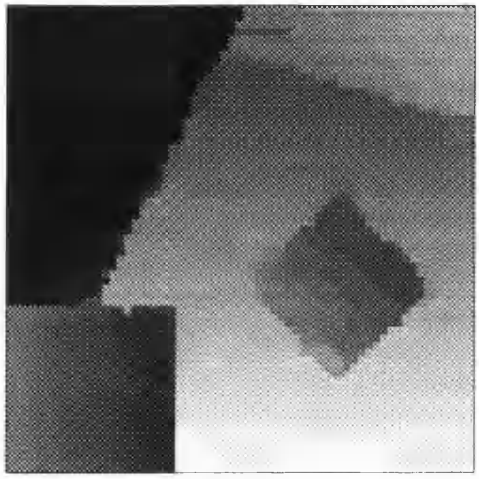
(a) First order, 4 neighbour, Δ=25 d=3



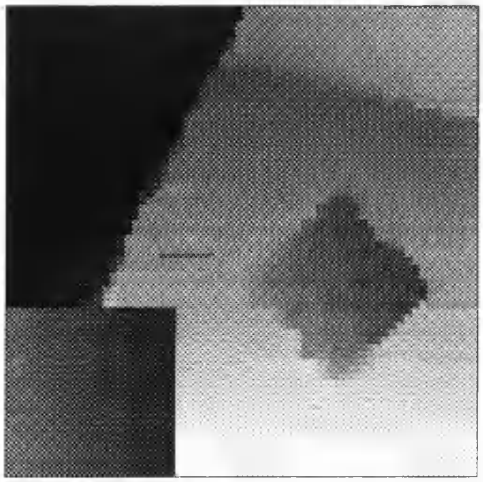
(b) First order, 8 neighbour, Δ=25 d=3



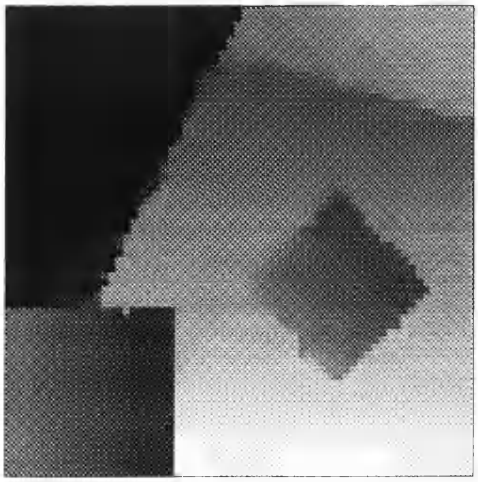
(c) Second order, 4 neighbour, Δ=10 d=3



(d) Second order, 8 neighbour, Δ=10 d=3



(e) Third order, 4 neighbour, Δ=10 d=3



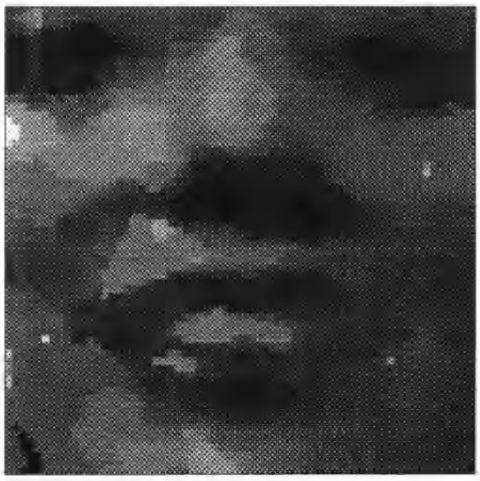
(f) Third order, 8 neighbour, Δ=10 d=3

Figure 3.5 Three levels of ICM reconstruction of a second order scene

(a) First order, 4 neighbour, Δ=40 d=3

(b) First order, 8 neighbour, Δ=40 d=3

(c) Second order, 4 neighbour, Δ=10 d=3



(d) Second order, 8 neighbour, Δ=10 d=3

(e) Third order, 4 neighbour, Δ=10 d=3

(f) Third order, 8 neighbour, Δ=10 d=3

Figure 3.6 Three levels of ICM reconstruction of a higher order scene

occurrence of outlying pixels in otherwise smooth regions (see, for example, the "keyhole" in Figure 3.4(b)). We believe that this behaviour is related to the parameter selection; an explanation is discussed in Section 3.4.5, together with a possible solution.

The six reconstructions of the second order scene are shown in Figure 3.5. Here the differences between the four and eight neighbour model reconstructions are less marked. In terms of recovering first order discontinuities, the eight neighbour model is possibly more successful, particularly with the central diagonally edged object. Notice that in the original scene, the edges of this object do not all form perfect diagonals, there are several translations, up or down, by a single pixel. One possible worry with selecting the parameters to recover diagonal edges is that diagonals may be fitted even when not appropriate. This does not appear to be the case here, and an attempt seems to have been made to recover the artifacts. The smoothing of any inadequately recovered discontinuities by higher order models is again apparent; see, for example, the left corner of the central object which appears to be smoothed into the background as the model order increases. As regards the order of the scene, the first order models are not fully able to recover the continuous gradation in the scene. The second order model improves on this, as would be expected. However, it is not until the third order reconstruction that the slopes are more clearly present. This may indicate a bad choice of Δ for the second order model; if Δ is too small, then the terraced effect produced by the first order reconstruction may be retained by the second order model which will view these discontinuities as genuine edges.

The higher order scene is possibly the hardest to reconstruct, in that it has not been generated by combining large third order regions of pixels. Given this, the third order reconstructions shown in Figure 3.6, are reasonable, particularly within the 64×44 strip of the grid for which there are recorded data values. The eight neighbour reconstructions are possibly less blurred than the four neighbour reconstructions. This is particularly apparent in the first order reconstructions. The effect of the different model orders is fairly clear, even though any areas of the scene which could be described as third order regions are generally quite small. In this case, any higher order smoothing of possible first order discontinuities is not a problem, since there would appear to be few genuine first order discontinuities in the original scene. Unfortunately, the eight neighbour model again exhibits the problem of isolated, outlying pixel values. The two images whose reconstructions exhibited the outliers, the first and the higher order scenes, will be reconstructed again in the next section, when we will consider this problem in more detail.

3.4.5 Further comments on the parameter selection

In the last section, Figures 3.4 and 3.6 demonstrated a difficulty with some restorations using the present model and parameter selection. This problem of outlying pixel values in the restorations tends to occur with heavily blurred, or very noisy records, and is obviously unacceptable. In the examples given by Geman and Reynolds, this phenomenon does not appear to occur. As an optimisation technique, they have used a modified Gibbs sampler based simulated annealing where the set of possible new pixel values is restricted to those lying within a certain small distance of either the current pixel value, or the record, or the current values of the adjacent pixels. This algorithm will be discussed in greater detail in Section 6.2.3. One result is that, in their implementation, these outliers would have virtually no chance of appearing. We have used standard single-site update ICM where at each pixel update, any value is permitted provided that it minimises the current penalty. To demonstrate why this might result in isolated outlying pixels, we present the following argument.

Suppose we were just to consider the likelihood contribution made to the energy by a single pixel s. We know the record formation from Equations (2.1) and (2.2), and we will assume that s is an interior pixel, so that records exist for both s, and a sufficiently large set of other pixels nearby in the grid.

$$\begin{aligned}
\lambda \sum_{t:s\in B_t} (\mathbf{Y}_t - (\mathbf{KX})_t)^2 &= \lambda \sum_{t:s\in B_t} (\mathbf{Y}_t - \sum_{r:r\in B_t} \gamma_{t-r}\mathbf{X}_r)^2 \\
&= \lambda \sum_{t:s\in B_t} (\mathbf{Y}_t - \gamma_{t-s}\mathbf{X}_s - \sum_{r:r\in B_t, r\neq s} \gamma_{t-r}\mathbf{X}_r)^2 \\
&= \lambda \sum_{t:s\in B_t} (\gamma_{t-s}\mathbf{X}_s - (\mathbf{Y}_t - \sum_{r:r\in B_t, r\neq s} \gamma_{t-r}\mathbf{X}_r))^2 \\
&= \lambda\{(\sum_{t:s\in B_t} \gamma_{t-s}^2)\mathbf{X}_s^2 - 2(\sum_{t:s\in B_t} \gamma_{t-s}(\mathbf{Y}_t - \sum_{r:r\in B_t, r\neq s} \gamma_{t-r}\mathbf{X}_r))\mathbf{X}_s + \\
&\qquad \sum_{t:s\in B_t} (\mathbf{Y}_t - \sum_{r:r\in B_t, r\neq s} \gamma_{t-r}\mathbf{X}_r)^2\}.
\end{aligned}$$

Suppose we then want to find the minimiser $\tilde{\mathbf{X}}_s$ of this quadratic in $\mathbf{X}_s$, holding $\mathbf{X}_{-s}$ fixed. Differentiating with respect to $\mathbf{X}_s$,

$$\tilde{\mathbf{X}}_s = \frac{\sum_{t:s\in B_t} \gamma_{t-s}(\mathbf{Y}_t - \sum_{r:r\in B_t, r\neq s} \gamma_{t-r}\mathbf{X}_r)}{\sum_{t:s\in B_t} \gamma_{t-s}^2}.$$

We can use the definitions of $\beta_s = \sum_{t\in S} \gamma_{t-s}^2$, and $Z_s(\eta) = \sum_{t\in S} \gamma_{t-s}\eta_t$, as given by Equation (3.19) to rewrite this expression for $\tilde{\mathbf{X}}_s$. In Section 2.1.1, we stated that $\gamma_{t-s}=0$ unless $s\in B_t$, the set of blurring neighbours of t. So, in these two

definitions, the summation over $t \in S$ can be replaced by one over $t: s \in B_t$. Also, since s is an internal pixel, it contributes to the maximum possible number of records, and β_s attains its maximum value β. Therefore, using $\mathbf{Y}_t = (\mathbf{K}\mathbf{X}^0)_t + \eta_t$,

$$\begin{aligned}
\tilde{\mathbf{X}}_s &= \frac{1}{\beta}\{ \sum_{t:s\in B_t} \gamma_{t-s}((\sum_{r:r\in B_t} \gamma_{t-r}\mathbf{X}_r^0 + \eta_t) - \sum_{r:r\in B_t, r\neq s} \gamma_{t-r}\mathbf{X}_r)\} \\
&= \frac{1}{\beta}\{ \sum_{t:s\in B_t} \gamma_{t-s}(\sum_{r:r\in B_t, r\neq s} \gamma_{t-r}(\mathbf{X}_r^0 - \mathbf{X}_r) + \gamma_{t-s}\mathbf{X}_s^0 + \eta_t)\} \\
&= \frac{1}{\beta}\{ \beta\mathbf{X}_s^0 + Z_s(\eta) + \sum_{t:s\in B_t} \gamma_{t-s} \sum_{r:r\in B_t, r\neq s} \gamma_{t-r}(\mathbf{X}_r^0 - \mathbf{X}_r)\}.
\end{aligned}$$

So the difference between the true value at the pixel s, $\mathbf{X}_s^0$, and the minimiser $\tilde{\mathbf{X}}_s$ of the likelihood contribution made by s (when the other pixel values are held fixed as $\mathbf{X}_{-s}$) can be written

$$\tilde{\mathbf{X}}_s - \mathbf{X}_s^0 = \frac{1}{\beta}\{ Z_s(\eta) + \sum_{t:s\in B_t} \sum_{r:r\in B_t, r\neq s} \gamma_{t-s}\gamma_{t-r}(\mathbf{X}_r^0 - \mathbf{X}_r)\}. \qquad (3.25)$$

The distribution of $Z_s(\eta)$ is known, $Z_s(\eta) \sim N(0, \beta\sigma^2)$, since we have assumed that each η_t is Normally distributed $N(0,\sigma^2)$. If we assume that we have correctly restored all the pixels around s, so $\mathbf{X}_{-s} = \mathbf{X}_{-s}^0$, and intuitively this might be thought to give us a good chance of correctly restoring s, then

$$\tilde{\mathbf{X}}_s - \mathbf{X}_s^0 \sim N(0, \frac{\sigma^2}{\beta}).$$

The constant β is equal to the sum of the blurring coefficients squared, and so will lie between 0 and 1. As the blurring becomes more severe, β will decrease. If there are n pixels involved in the blurring of a pixel s, $|B_s| = n$, then β will lie between $1/n$ and 1, with the lower limit attained by uniform blurring. So the variance of this possible difference could become very large when either the variance of the noise becomes high, or the blurring becomes severe.

So far, we have only considered the likelihood contribution to the energy. There will also be a prior contribution which should be counteracting any tendency for $\mathbf{X}_s$ to stray too far from the values of its Markov random field neighbours. The $\varphi(u)$ used for our reconstructions has been selected in part for the fact that it has a finite asymptotic limit as $u \to \infty$. Its strict concavity implies that, after some value of u, the penalties for increasing u can only be very gradually increasing. This should ensure that genuine edges stand a reasonable chance of being recovered. However, it may also prevent the suppression of extreme values at pixels with a large value of $\tilde{\mathbf{X}}_s - \mathbf{X}_s^0$. If this is the case, then there are at least two ways in which we might be able to avoid this problem.

The first potential solution is to increase the scaling parameter Δ. Since Δ controls the spread of $\varphi(u)$, increasing Δ will increase the value of u after which the penalty becomes roughly constant. This effect could be seen in Figure 3.1, which plotted $\varphi(\,)$ for three different values of Δ. An example of the effect in reconstruction of increasing Δ is shown in Figure 3.7. This gives six ICM reconstructions of the third order face image used in the previous section. The values of all the settings other than Δ are kept constant, while Δ is increased in steps of 10 from 25 to 75 (the example in Figure 3.6 used Δ=40). The extreme outlier problem does appear to be suppressed by the increases, although for the larger values of Δ, there may be a slight deterioration in the deblurring. This is not unexpected given the interpretation of Δ as a scale to determining when a discontinuity is genuinely an edge.

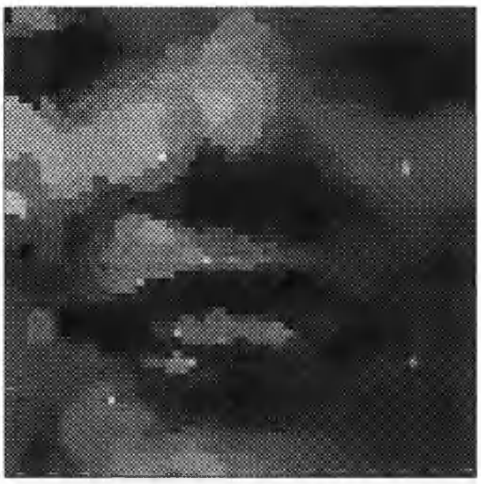
The other possible solution to the problem is to increase d. Equation (3.23) defines the parameter d to be the $1-\varepsilon/(2|S^0|)$ quantile of a standard Normal. Its significance is that this represents the confidence with which, under all the various constraints, the conditions for a diagonal step edge to be a coordinate-wise minimum of the energy are not broken at a particular pixel. In Section 3.4.2, we showed that this condition could be rephrased in terms of a condition on $Z_s(\eta)$ (Equation (3.21)), and so from Equation (3.25), when $\underline{\mathbf{X}}_{-s}=\underline{\mathbf{X}}^0_{-s}$,

$$\frac{2}{\sqrt{2\pi}}\int_0^d e^{-t^2/2}dt \le P_\sigma(|Z_s(\eta)| < L_s^i)$$

$$= P_\sigma(|\tilde{\mathbf{X}}_s - \mathbf{X}_s^0| < \frac{L_s^i}{\beta}).$$

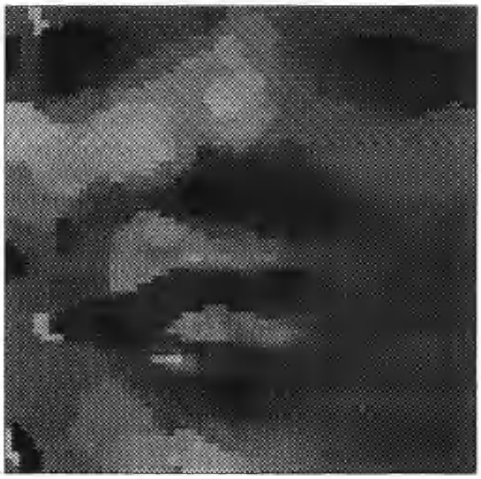
Increasing d, increases the probability that this inequality is satisfied. Figure 3.8 shows six ICM reconstructions of the first order scene used in the previous section, for increasing values of d. The original settings of all parameters are retained, except for d which is increased in steps of 1 from 0.5 to 5.5 (the example in Figure 3.4 used d=3). Again the outlier problem appears to be suppressed by increasing d, although the drawback in this case seems to be a tendency to overstraighten edges (oversmoothing). This is not surprising when the role of d in the expression for the smoothing parameter λ, Equation (3.24), is considered; increasing d, decreases λ. The drawbacks of increasing either Δ or d could possibly be reduced by using a smaller, simultaneous increase in both parameters.

The outlier phenomenon discussed will probably not occur when a modified algorithm such as that used by Geman and Reynolds is employed. However with a standard algorithm such as ICM, our recommendation is that in cases where the noise variance is high or the blurring severe, it may be advisable to select larger values of Δ or d than might be used with a less degraded record.



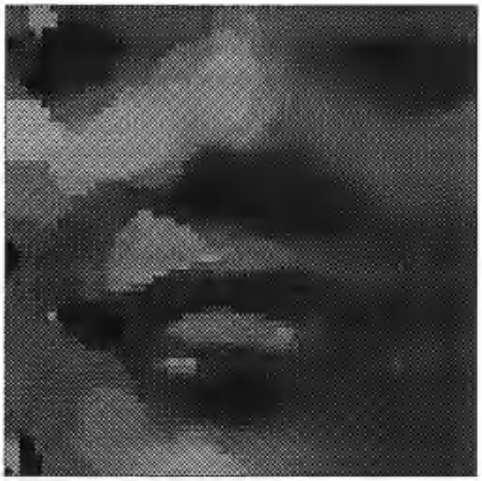
(a) Δ=25

(b) Δ=35



(c) Δ=45

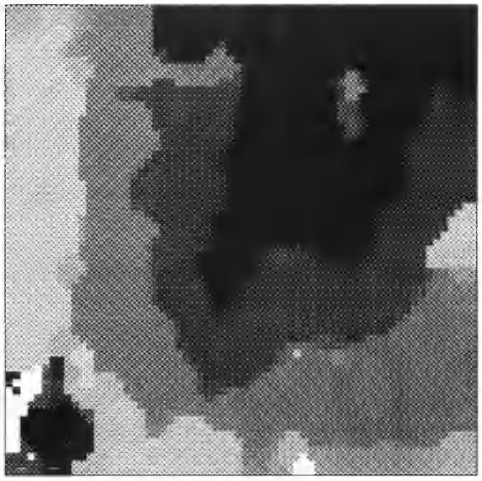
(d) Δ=55



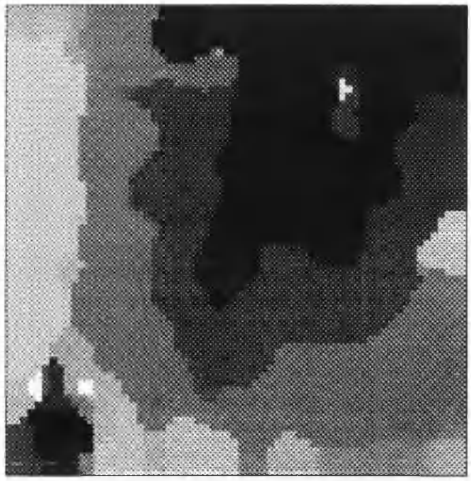
(e) Δ=65

(f) Δ=75

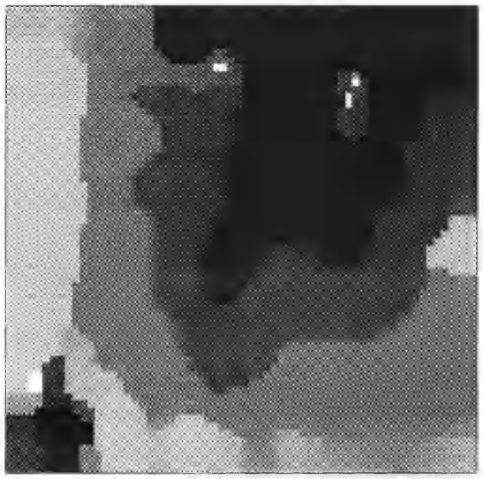
Figure 3.7 Six eight neighbour first order ICM reconstructions with increasing values of Δ and fixed d=3



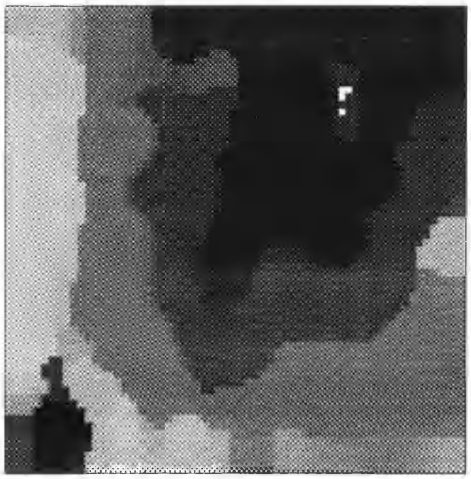
(a) d=0.5



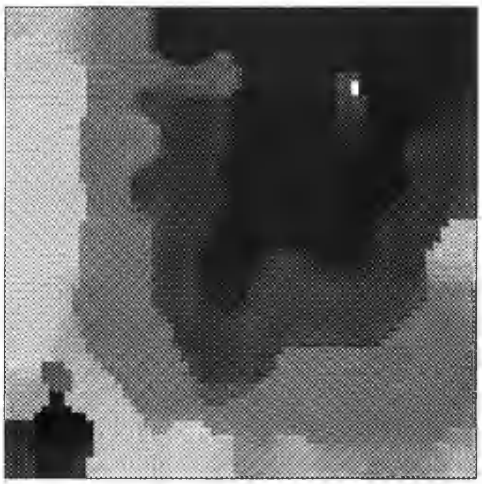
(b) d=1.5



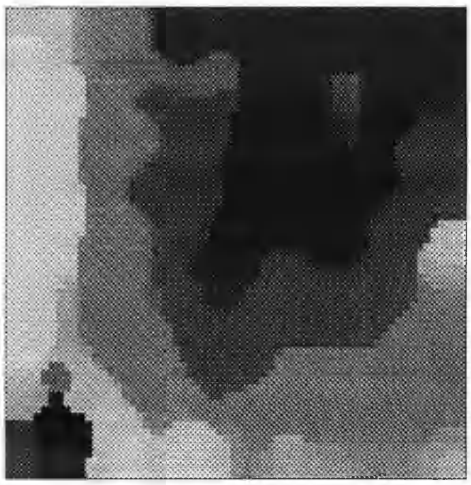
(c) d=2.5



(d) d=3.5



(e) d=4.5



(f) d=5.5

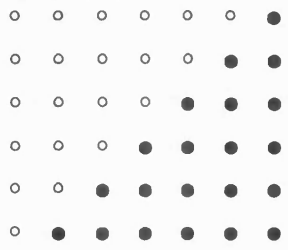
Figure 3.8 Six eight neighbour first order ICM reconstructions with increasing values of d and fixed Δ=15

3.5 Appendix

3.5.1 Introduction

In Section 3.4.1, we defined a coordinate-wise minimum of the prior $\Phi(\mathbf{X})$, and stated that a diagonal step edge was such a minimum when working with our extended eight neighbour model. This appendix provides the constants required for that result, for the three orders of the model. The notation used here has been defined in Section 3.4.1.

Let ∘ denote a pixel taking the value J ($J>0$), and • denote a pixel taking the value 0. Then, we are interested in a diagonal step edge $\mathbf{X}^0$ of the form



The values of the pixels appear in the prior through paired differences, so the choice of 0 and J is without loss of generality for any edge of size J. For notational simplicity, we will take the parameter $\Delta=1$. Again this is without loss of generality, since the value J could be taken to absorb Δ.

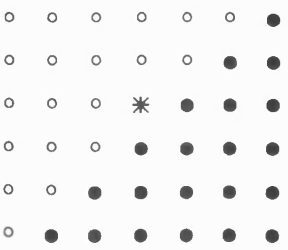
The quantity in which we are interested is the change in the prior when one pixel s is perturbed in value by an amount u,

$$f_s^i(u) = \Phi(\mathbf{X}^0 + \mathbf{u}^s) - \Phi(\mathbf{X}^0)$$

$$= \sum_{c:s\in c} w_c^i\{\varphi(D_c^i(\mathbf{X}^0+\mathbf{u}^s)) - \varphi(D_c^i(\mathbf{X}^0))\}$$

where $\mathbf{u}^s$ denotes the vector taking the value 0 at all coordinates except the s^{th} where it takes the value u, and the weights $\{w_c^i\}$ are given by Equations (3.9), (3.10) and (3.11). By definition, a diagonal step edge is a coordinate-wise minimum of the prior if $f_s^i(u)>0$, $\forall s\in S^0$, $u\neq 0$.

We will assume that the pixel s to be perturbed is a ∘ pixel on the boundary between the regions. Geman and Reynolds show, for the first order case, that boundary pixels such as these are the least stable (have the lowest $f_s^i(u)$). In this case, the pixel's value is perturbed from J to $J+u$; it is denoted below by ✳.



The results in the next two sections for the first and second order models are true for any $\varphi(u)$ satisfying the conditions that it is even, concave and increasing on the positive half-line, with $\varphi(0)=0$. We will require the following lemma (Geman and Reynolds' Lemma 1):

$$\varphi(J) \leq \varphi(u) + \varphi(J-u), \forall u, J>0. \tag{3.26}$$

This is proved for $u \geq J$ or $u \leq 0$ by noting that $\varphi(u)$ is even and increasing on the positive half line, with $\varphi(0)=0$. For $0<u<J$, we will use the concavity of $\varphi(u)$. Approximating the derivative of $\varphi(u)$, this implies that for any three positive numbers $0<a \leq b<c$,

$$\frac{\varphi(a)-\varphi(0)}{a-0} \geq \frac{\varphi(c)-\varphi(b)}{c-b}, \quad \text{with } \varphi(0)=0.$$

If $0<u \leq J-u<J$, apply this inequality with $a=u$, $b=J-u$ and $c=J$. If $0<J-u \leq u<J$, apply this inequality with $a=J-u$, $b=u$ and $c=J$.

3.5.2 First order model

Consider the eight first order cliques defined in Section 3.3 involving the pixel s. Using the down-weighting given in Equation (3.9),

$$\sum_{c:s \in c} w_c \varphi(D_c^1(\mathbf{X}^0)) = 2\,(\varphi(0) + \varphi(J)) + \frac{1}{\sqrt{2}}\,(3\varphi(0) + \varphi(J)),$$

$$\sum_{c:s \in c} w_c \varphi(D_c^1(\mathbf{X}^0+\mathbf{u}^s)) = 2\,(\,\varphi(u) + \varphi(J+u)\,) + \frac{1}{\sqrt{2}}\,(\,3\,\varphi(u) + \varphi(J+u)\,).$$

Therefore, using the fact that $\varphi(0)=0$,

$$f_s^1(u) = 2\,(\varphi(u)+\varphi(J+u)-\varphi(J)) + \frac{1}{\sqrt{2}}\,(3\varphi(u)+\varphi(J+u)-\varphi(J)).$$

Setting u to $-u$ in Equation (3.26) implies that $\varphi(J) \leq \varphi(u)+\varphi(J+u)$, since $\varphi(u)$ is even. This gives us the inequality

$$f_s^1(u) \geq \frac{2}{\sqrt{2}}\,\varphi(u) \quad \forall\, u, \forall\, s \in S^0. \tag{3.27}$$

Since $\varphi(u)>0$ for non-zero u, this shows that the diagonal step edge is a coordinate-wise minimum of the prior for the first order model incorporating diagonal neighbours.

3.5.3 Second order model

The approach for the second order model is identical to that for the first order model. Consider the thirty-six second order cliques defined in Section 3.3 involving the pixel s. Using the down-weighting given in Equation (3.10),

$$\sum_{c:s \in c} w_c \varphi(D_c^2(\mathbf{X}^0)) = (3\varphi(0)+7\varphi(J)) + \frac{1}{2}\,(8\varphi(0)+2\varphi(J)) + \frac{1}{\sqrt{2}}\,(12\varphi(0)+4\varphi(J)),$$

$$\sum_{c:s\in c} w_c\varphi(D_c^2(\mathbf{X}^0+\mathbf{u}^s)) = (3\varphi(u)+2\varphi(J+2u)+5\varphi(J+u)) + \frac{1}{2}\,(7\varphi(u)+\varphi(J+2u)+$$

$$\varphi(2u)+\varphi(J+u)) + \frac{1}{\sqrt{2}}\,(12\varphi(u)+4\varphi(J+u)).$$

Therefore, using the fact that $\varphi(0)=0$,

$$f_s^2(u) = (\frac{13}{2}+\frac{12}{\sqrt{2}})\,\varphi(u) + \frac{5}{2}\,\varphi(J+2u) + (\frac{11}{2}+\frac{4}{\sqrt{2}})\,\varphi(J+u) +$$

$$\frac{1}{2}\,\varphi(2u) - (8+\frac{4}{\sqrt{2}})\,\varphi(J).$$

Setting u to $-u$ in Equation (3.26) implies that $\varphi(J)\le\varphi(u)+\varphi(J+u)$, so

$$f_s^2(u) \ge (1+\frac{8}{\sqrt{2}})\,\varphi(u) + \frac{5}{2}\,\varphi(J+2u) + \frac{1}{2}\,\varphi(2u) - \frac{5}{2}\,\varphi(J).$$

Then setting u to $-2u$ in Equation (3.26) implies that $\varphi(J)\le\varphi(2u)+\varphi(J+2u)$, so

$$f_s^2(u) \ge (1+\frac{8}{\sqrt{2}})\,\varphi(u) - 2\,\varphi(2u).$$

Finally, using the concavity of $\varphi(\,)$, and the fact that $\varphi(0)=0$, which imply that $\frac{\varphi(2u)}{2u}\le\frac{\varphi(u)}{u}$ for $u\neq 0$,

$$f_s^2(u) \ge (\frac{8}{\sqrt{2}}-3)\,\varphi(u) \quad \forall\, u,\ \forall\, s\in S^0. \tag{3.28}$$

Since $\varphi(u)>0$ for non-zero u, this shows that the diagonal step edge is a coordinate-wise minimum of the prior for the second order model incorporating diagonal neighbours.

3.5.4 Third order model

The approach for the third order model must be modified from that used for the first and second order models. However, as before, we will begin by defining $f_s^3(u)$. Consider the eighty-eight third order cliques defined in Section 3.3 involving the pixel s. Using the down-weighting given in Equation (3.11),

$$\sum_{c:s\in c} w_c\varphi(D_c^3(\mathbf{X}^0)) = (4\varphi(0)+10\varphi(J)+6\varphi(2J)) + \frac{1}{2}\,(18\varphi(0)+4\varphi(J)+2\varphi(2J)) +$$

$$\frac{1}{\sqrt{2}}\,(14\varphi(0)+10\varphi(J)) + \frac{1}{2\sqrt{2}}\,(17\varphi(0)+2\varphi(J)+\varphi(2J)).$$

$$\sum_{c:s\in c} w_c\varphi(D_c^3(\mathbf{X}^0+\mathbf{u}^s)) = (6\varphi(J+u)+4\varphi(u)+2\varphi(2(J+u))+2\varphi(J+2u)+2\varphi(2J+u)+$$

$$2\varphi(2J+3u)+2\varphi(J+3u)) + \frac{1}{2}\,(4\varphi(J+u)+2\varphi(2(J+u))+$$

$$12\varphi(u)+6\varphi(2u)) + \frac{1}{\sqrt{2}}\,(4\varphi(J+u)+10\varphi(u)+4\varphi(J+2u)+$$

$$4\varphi(2u)+2\varphi(J-u)) + \frac{1}{2\sqrt{2}}\,(\varphi(J+u)+11\varphi(u)+4\varphi(2u)+$$

$$2\varphi(3u)+\varphi(2J+3u)+\varphi(J+3u)).$$

These expressions are far more complicated than the corresponding lower order expressions, reflecting the growth in the total number of cliques. We will write $f_s^3(u)$ in such a way as to emphasize that it is the difference of the two penalties,

$$f_s^3(u) = (8+\frac{9}{2\sqrt{2}})(\varphi(J+u)-\varphi(J)) + 2(\varphi(2J+u)-\varphi(2J)) + \frac{2}{\sqrt{2}}(\varphi(J-u)-\varphi(J)) +$$

$$(10+\frac{31}{2\sqrt{2}})\varphi(u) + 3(\varphi(2(J+u))-\varphi(2J)) + (2+\frac{4}{\sqrt{2}})(\varphi(J+2u)-\varphi(J)) +$$

$$(2+\frac{1}{2\sqrt{2}})(\varphi(2J+3u)-\varphi(2J)) + (2+\frac{1}{2\sqrt{2}})(\varphi(J+3u)-\varphi(J)) +$$

$$(3+\frac{6}{\sqrt{2}})\varphi(2u) + \frac{1}{\sqrt{2}}\varphi(3u), \quad \text{since } \varphi(0)=0.$$

It might be expected from the geometry of the diagonal step edge that the penalty would increase if $u>0$ or $u\leq -J$ since the $*$ pixel is then moving further away from both sets of unperturbed pixel. This is indeed the case:

(i) $u > 0$: $J+u \geq J \Rightarrow \varphi(J+u) \geq \varphi(J)$

$2J+u \geq 2J \Rightarrow \varphi(2J+u) \geq \varphi(2J)$

$2(J+u) \geq 2J \Rightarrow \varphi(2(J+u)) \geq \varphi(2J)$

$J+2u \geq J \Rightarrow \varphi(J+2u) \geq \varphi(J)$

$2J+3u \geq 2J \Rightarrow \varphi(2J+3u) \geq \varphi(2J)$

$J+3u \geq J \Rightarrow \varphi(J+3u) \geq \varphi(J)$

Equation (3.26) $\Rightarrow \varphi(J-u)-\varphi(J) \geq -\varphi(u)$

$3u \geq 2u \geq u \Rightarrow \varphi(3u) \geq \varphi(2u) \geq \varphi(u)$

$$\Rightarrow \quad f_s^3(u) \geq (13 + \frac{41}{2\sqrt{2}})\,\varphi(u) \geq 0, \quad u > 0. \tag{3.29}$$

(ii) $u < -J$: $J-u \geq J \Rightarrow \varphi(J-u) \geq \varphi(J)$

$-(J+2u) \geq J \Rightarrow \varphi(J+2u) \geq \varphi(2J)$

$-(J+3u) \geq J \Rightarrow \varphi(J+3u) \geq \varphi(J)$

Equation (3.26) $\Rightarrow \varphi(J+u)-\varphi(J) \geq -\varphi(u)$ (setting u to $-u$)

Equation (3.26) $\Rightarrow \varphi(2J+u)-\varphi(2J) \geq -\varphi(u)$ (setting u to $-u$, J to $2J$)

Equation (3.26) $\Rightarrow \varphi(2(J+u))-\varphi(2J) \geq -\varphi(2u)$ (setting u to $-2u$, J to $2J$)

Equation (3.26) $\Rightarrow$ $\varphi(2J+3u)-\varphi(2J) \geq -\varphi(3u)$ (setting u to $-3u$, J to $2J$)

So, $f_s^3(u) \geq \frac{11}{\sqrt{2}}\,\varphi(u) + \frac{6}{\sqrt{2}}\,\varphi(2u) - (2-\frac{1}{2\sqrt{2}})\,\varphi(3u)$.

But, again from the concavity of $\varphi(\,)$ and the fact that $\varphi(0)=0$, $\frac{\varphi(3u)}{3u} \leq \frac{\varphi(2u)}{2u}$ for $u \neq 0$. So replacing $\varphi(3u)$ in the above expression by $\frac{3}{2}\,\varphi(2u)$ gives $f_s^3(u) \geq \frac{11}{\sqrt{2}}\,\varphi(u) + (\frac{27}{4\sqrt{2}}-3)\,\varphi(2u)$. Then using $\varphi(2u) \geq \varphi(u)$,

$$\Rightarrow \qquad f_s^3(u) \geq (\,\frac{71}{4\sqrt{2}} - 3\,)\,\varphi(u) \geq 0, \quad u \leq -J. \tag{3.30}$$

Unfortunately, when $0>u>-J$, it is not possible to use this approach to show that $f_s^3(u)$ is strictly non-negative. Even working with the specific $\varphi(\,)$ given in Equation (3.5), for which there might be a tighter bound than Equation (3.26), there are some values of u for which $f_s^3(u)<0$, unless J is constrained. This problem was identified by a numerical evaluation of $f_s^3(u)$ for various J. The approach taken here is to consider subintervals over which each of the paired expressions in $f_s^3(u)$ is differentiable. Then, for each paired difference, we can find a lower limit in terms of multiples of $\varphi(u)$. We do this by minimising each difference individually, since the simultaneous minimisation of the whole expression proved intractable. These limits will be a function of J. The arbitrary choice of $J \geq 1$ has then been made, and under this condition a positive limit for $f_s^3(u)/\varphi(u)$ can be found. This involves separately minimising the positive terms, and maximising the negative terms, over J in $[1,\infty)$. It should be remembered that we have taken $\Delta=1$, more generally, the condition which we have imposed requires that $J/\Delta>1$.

(iii) $0 > u \geq \frac{-J}{3}$:

$$\frac{\varphi(J+u)-\varphi(J)}{\varphi(u)} = \frac{-(1-u)}{(1+J)(1+J+u)} \geq \frac{-(1+J/3)}{(1+J)(1+2J/3)}$$

$$\frac{\varphi(2J+u)-\varphi(2J)}{\varphi(u)} = \frac{-(1-u)}{(1+2J)(1+2J+u)} \geq \frac{-(1+J/3)}{(1+2J)(1+5J/3)}$$

$$\frac{\varphi(J-u)-\varphi(J)}{\varphi(u)} = \frac{(1-u)}{(1+J)(1+J-u)} \geq \frac{1}{(1+J)(1+J)}$$

$$\frac{\varphi(2(J+u))-\varphi(2J)}{\varphi(u)} = \frac{-2(1-u)}{(1+2J)(1+2J+2u)} \geq \frac{-2(1+J/3)}{(1+2J)(1+4J/3)}$$

$$\frac{\varphi(J+2u)-\varphi(J)}{\varphi(u)} = \frac{-2(1-u)}{(1+J)(1+J+2u)} \geq \frac{-2}{(1+J)}$$

$$\frac{\varphi(2J+3u)-\varphi(2J)}{\varphi(u)} = \frac{-3(1-u)}{(1+2J)(1+2J+3u)} \geq \frac{-3(1+J/3)}{(1+2J)(1+J)}$$

$$\frac{\varphi(J+3u)-\varphi(J)}{\varphi(u)} = \frac{-3(1-u)}{(1+J)(1+J+3u)} \geq \frac{-3(1+J/3)}{(1+J)}$$

$$\frac{\varphi(2u)}{\varphi(u)} = \frac{2(1-u)}{(1-2u)} \geq \frac{2(1+J/3)}{(1+2J/3)}$$
$$\frac{\varphi(3u)}{\varphi(u)} = \frac{3(1-u)}{(1-3u)} \geq \frac{3(1+J/3)}{(1+J)}$$

So, putting all these inequalities together into the expression for $f_s^3(u)/\varphi(u)$ gives

$$\frac{f_s^3(u)}{\varphi(u)} \geq (10+\frac{31}{2\sqrt{2}}) - (8+\frac{9}{2\sqrt{2}})\frac{(1+J/3)}{(1+J)(1+2J/3)} - 2\frac{(1+J/3)}{(1+2J)(1+5J/3)} +$$
$$\frac{2/\sqrt{2}}{(1+J)(1+J)} - 6\frac{(1+J/3)}{(1+2J)(1+4J/3)} - \frac{2(2+4/\sqrt{2})}{(1+J)} + (6+\frac{12}{\sqrt{2}})\frac{(1+J/3)}{(1+2J/3)} -$$
$$(6+\frac{3}{2\sqrt{2}})\frac{(1+J/3)}{(1+J)(1+2J)} - (6-\frac{3}{2\sqrt{2}})\frac{(1+J/3)}{(1+J)}$$
$$\geq \frac{104}{105} + \frac{491}{30\sqrt{2}} \quad \text{when } J \geq 1.$$

$$\Rightarrow \qquad f_s^3(u) \geq \left(\frac{104}{105} + \frac{491}{30\sqrt{2}} \right) \varphi(u), \quad 0 > u \geq -\frac{J}{3}, \; J \geq 1. \tag{3.31}$$

(*iv*) $\frac{-J}{3} > u \geq \frac{-J}{2}$:
$$\frac{\varphi(J+u)-\varphi(J)}{\varphi(u)} = \frac{-(1-u)}{(1+J)(1+J+u)} \geq \frac{-1}{(1+J)}$$
$$\frac{\varphi(2J+u)-\varphi(2J)}{\varphi(u)} = \frac{-(1-u)}{(1+2J)(1+2J+u)} \geq \frac{-(1+J/2)}{(1+2J)(1+3J/2)}$$
$$\frac{\varphi(J-u)-\varphi(J)}{\varphi(u)} = \frac{(1-u)}{(1+J)(1+J-u)} \geq \frac{(1+J/3)}{(1+J)(1+4J/3)}$$
$$\frac{\varphi(2(J+u))-\varphi(2J)}{\varphi(u)} = \frac{-2(1-u)}{(1+2J)(1+2J+2u)} \geq \frac{-2(1+J/2)}{(1+2J)(1+J)}$$
$$\frac{\varphi(J+2u)-\varphi(J)}{\varphi(u)} = \frac{-2(1-u)}{(1+J)(1+J+2u)} \geq \frac{-2(1+J/2)}{(1+J)}$$
$$\frac{\varphi(2J+3u)-\varphi(2J)}{\varphi(u)} = \frac{-3(1-u)}{(1+2J)(1+2J+3u)} \geq \frac{-3}{(1+2J)}$$
$$\frac{\varphi(J+3u)-\varphi(J)}{\varphi(u)} = \frac{-(2J+3u)(1-u)}{(1+J)(1+J+3u)(-u)} \geq \frac{-3(1+J/3)}{(1+J)}$$
$$\frac{\varphi(2u)}{\varphi(u)} = \frac{2(1-u)}{(1-2u)} \geq \frac{2(1+J/2)}{(1+J)}$$
$$\frac{\varphi(3u)}{\varphi(u)} = \frac{3(1-u)}{(1-3u)} \geq \frac{3(1+J/2)}{(1+3J/2)}$$

So, putting all these inequalities together into the expression for $f_s^3(u)/\varphi(u)$ gives

$$\frac{f_s^3(u)}{\varphi(u)} \geq (10+\frac{31}{2\sqrt{2}}) - \frac{8+9/(2\sqrt{2})}{(1+J)} - 2\frac{(1+J/2)}{(1+2J)(1+3J/2)} + \frac{2}{\sqrt{2}}\frac{(1+J/3)}{(1+J)(1+4J/3)} -$$
$$6\frac{(1+J/2)}{(1+2J)(1+J)} + (2+\frac{4}{\sqrt{2}})\frac{(1+J/2)}{(1+J)} - \frac{3(2+1/(2\sqrt{2}))}{(1+2J)} -$$

$$3(2+\frac{1}{2\sqrt{2}})\frac{(1+J/3)}{(1+J)} + \frac{3}{\sqrt{2}}\frac{(1+J/2)}{(1+3J/2)}$$

$$\geq -\frac{9}{10} + \frac{59}{4\sqrt{2}} \qquad \text{when } J \geq 1.$$

$$\Rightarrow \qquad f_s^3(u) \geq \left(-\frac{9}{10} + \frac{59}{4\sqrt{2}} \right) \varphi(u), \quad -\frac{J}{3} > u \geq -\frac{J}{2}, \; J \geq 1. \tag{3.32}$$

(v) $\dfrac{-J}{2} > u \geq \dfrac{-2J}{3}$:

$$\frac{\varphi(J+u)-\varphi(J)}{\varphi(u)} = \frac{-(1-u)}{(1+J)(1+J+u)} \geq \frac{-(1+2J/3)}{(1+J)(1+J/3)}$$

$$\frac{\varphi(2J+u)-\varphi(2J)}{\varphi(u)} = \frac{-(1-u)}{(1+2J)(1+2J+u)} \geq \frac{-(1+2J/3)}{(1+2J)(1+4J/3)}$$

$$\frac{\varphi(J-u)-\varphi(J)}{\varphi(u)} = \frac{(1-u)}{(1+J)(1+J-u)} \geq \frac{(1+J/2)}{(1+J)(1+3J/2)}$$

$$\frac{\varphi(2(J+u))-\varphi(2J)}{\varphi(u)} = \frac{-2(1-u)}{(1+2J)(1+2J+2u)} \geq \frac{-2}{(1+2J)}$$

$$\frac{\varphi(J+2u)-\varphi(J)}{\varphi(u)} = \frac{-2(1-u)(J+u)}{(1+J)(1-J-2u)(-u)} \geq \frac{-2(1+J/2)}{(1+J)}$$

$$\frac{\varphi(2J+3u)-\varphi(2J)}{\varphi(u)} = \frac{-3(1-u)}{(1+2J)(1+2J+3u)} \geq \frac{-3(1+2J/3)}{(1+2J)}$$

$$\frac{\varphi(J+3u)-\varphi(J)}{\varphi(u)} = \frac{-(2J+3u)(1-u)}{(1+J)(1+J+3u)(-u)} \geq \frac{-1}{(1+J)}$$

$$\frac{\varphi(2u)}{\varphi(u)} = \frac{2(1-u)}{(1-2u)} \geq \frac{2(1+2J/3)}{(1+4J/3)}$$

$$\frac{\varphi(3u)}{\varphi(u)} = \frac{3(1-u)}{(1-3u)} \geq \frac{3(1+2J/3)}{(1+2J)}$$

So, putting all these inequalities together into the expression for $f_s^3(u)/\varphi(u)$ gives

$$\frac{f_s^3(u)}{\varphi(u)} \geq (10+\frac{31}{2\sqrt{2}}) - (8+\frac{9}{2\sqrt{2}})\frac{(1+2J/3)}{(1+J)(1+J/3)} - 2\frac{(1+2J/3)}{(1+2J)(1+4J/3)} +$$

$$\frac{2}{\sqrt{2}}\frac{(1+J/2)}{(1+J)(1+3J/2)} - \frac{6}{(1+2J)} - 2(2+\frac{4}{\sqrt{2}})\frac{(1+J/2)}{(1+J)} +$$

$$2(3+\frac{6}{\sqrt{2}})\frac{(1+2J/3)}{(1+4J/3)} - 3(2-\frac{1}{2\sqrt{2}})\frac{(1+2J/3)}{(1+2J)} - \frac{(2+1/(2\sqrt{2}))}{(1+J)}$$

$$\geq -\frac{38}{21} + \frac{637}{48\sqrt{2}} \qquad \text{when } J \geq 1.$$

$$\Rightarrow \qquad f_s^3(u) \geq \left(-\frac{38}{21} + \frac{637}{48\sqrt{2}} \right), \quad -\frac{J}{2} > u \geq -\frac{2J}{3}, \; J \geq 1. \tag{3.33}$$

(vi) $\dfrac{-2J}{3} > u \geq -J$:

$$\frac{\varphi(J+u)-\varphi(J)}{\varphi(u)} = \frac{-(1-u)}{(1+J)(1+J+u)} \geq -1$$

$$\frac{\varphi(2J+u)-\varphi(2J)}{\varphi(u)} = \frac{-(1-u)}{(1+2J)(1+2J+u)} \geq \frac{-1}{(1+2J)}$$

$$\frac{\varphi(J-u)-\varphi(J)}{\varphi(u)} = \frac{(1-u)}{(1+J)(1+J-u)} \geq \frac{(1+2J/3)}{(1+J)(1+5J/3)}$$

$$\frac{\varphi(2(J+u))-\varphi(2J)}{\varphi(u)} = \frac{-2(1-u)}{(1+2J)(1+2J+2u)} \geq \frac{-2(1+J)}{(1+2J)}$$

$$\frac{\varphi(J+2u)-\varphi(J)}{\varphi(u)} = \frac{-2(1-u)(J+u)}{(1+J)(1-J-2u)(-u)} \geq \frac{-(1+2J/3)}{(1+J)(1+J/3)}$$

$$\frac{\varphi(2J+3u)-\varphi(2J)}{\varphi(u)} = \frac{-(4J+3u)(1-u)}{(1+2J)(1-2J-3u)(-u)} \geq \frac{-3(1+2J/3)}{(1+2J)}$$

$$\frac{\varphi(J+3u)-\varphi(J)}{\varphi(u)} = \frac{-(2J+3u)(1-u)}{(1+J)(1+J+3u)(-u)} \geq 0$$

$$\frac{\varphi(2u)}{\varphi(u)} = \frac{2(1-u)}{(1-2u)} \geq \frac{2(1+J)}{(1+2J)}$$

$$\frac{\varphi(3u)}{\varphi(u)} = \frac{3(1-u)}{(1-3u)} \geq \frac{3(1+J)}{(1+3J)}$$

So, putting all these inequalities together into the expression for $f_s^3(u)/\varphi(u)$ gives

$$\frac{f_s^3(u)}{\varphi(u)} \geq (10+\frac{31}{2\sqrt{2}}) - (8+\frac{9}{2\sqrt{2}}) - \frac{2}{(1+2J)} + \frac{2}{\sqrt{2}}\frac{(1+2J/3)}{(1+J)(1+5J/3)} +$$

$$\frac{12}{\sqrt{2}}\frac{(1+J)}{(1+2J)} - (2+\frac{4}{\sqrt{2}})\frac{(1+2J/3)}{(1+J)(1+J/3)} - 3(2+\frac{1}{2\sqrt{2}})\frac{(1+2J/3)}{(1+2J)} +$$

$$\frac{3}{\sqrt{2}}\frac{(1+J)}{(1+3J)}$$

$$\geq -\frac{13}{4} + \frac{44}{3\sqrt{2}} \qquad \text{when } J \geq 1.$$

$$\Rightarrow \qquad f_s^3(u) \geq (-\frac{13}{4} + \frac{44}{3\sqrt{2}}), \quad -\frac{2J}{3} > u \geq -J, \; J \geq 1. \tag{3.34}$$

The whole range of non-zero u has now been covered with positive bounds for $f_s^3(u)$ in terms of $\varphi(u)$, under the condition that $J \geq 1$. By comparing the bounds in Equations (3.29) to (3.34), the minimum bound can be seen to occur for $\frac{-2J}{3} > u \geq -J$.

$$f_s^3(u) \geq (-\frac{13}{4} + \frac{44}{3\sqrt{2}}) \varphi(u) , \quad \forall u \neq 0, \; \forall s \in S^0, \; J \geq 1. \tag{3.35}$$

Since $\varphi(u)>0$ for non-zero u, this shows that the diagonal step edge is a coordinate-wise minimum of the prior for the third order model including diagonal neighbours, provided that $J/\Delta \geq 1$. The bound given in Equation (3.35) is only applicable to our specific $\varphi(\,)$. We note that it should be possible to find a better limit by some numerical approach.

Chapter 4: Multiple-Site Update Methods

4.1 The need for multiple-site updates

4.1.1 Introduction

In Sections 2.2 and 2.4.1, we described the minimisation techniques simulated annealing and ICM. These two procedures are generally implemented with single-site updating. That is, we select a sequence in which the pixels will be updated; usually this is chosen to be a raster scan ordering of the pixel grid. Then, in updating the currently selected pixel s, we fix all the other pixel values, $\mathbf{x}_{-s}$, and apply either algorithm to generate the next value for $\mathbf{X}_s$. In the next two sections, we will describe some of the possible difficulties which can arise when using single-site ICM or simulated annealing, and also indicate how updating groups of pixels simultaneously might be advantageous.

Computational complexity precludes some of the more naive attempts at multiple-site updating; with ICM, there may be too many possible configurations over which to search. In the simulated annealing case, the proposal distribution may become too complex to sample efficiently. The purpose of this chapter is to provide a brief review of some of the existing approaches to multiple-site updating. Some of these methods, for example the Swendsen and Wang algorithm, originally arose in the statistical physics literature concerned with the simulation of large systems of interacting bodies. Approaches such as these are intended to provide improved sampling algorithms, and so could also be incorporated into simulated annealing. Other methods, such as the cascade algorithm or the renormalised group approach, attempt to simplify the original problem by generating a sequence of related minimisation problems to solve; each problem in the sequence could be tackled with either simulated annealing or ICM. The methods described in this chapter vary in the degree to which they are readily implementable.

4.1.2 Problems with ICM

ICM is a deterministic, strictly downhill minimisation procedure. In a raster scan single-site version, taking each pixel s in turn, $\mathbf{x}_s$ is updated to minimise the current energy, holding the value of all other pixels fixed. The algorithm cannot escape from local minima, and is heavily dependent on the initial starting value. The quality of the restoration, and the optimality of the local minimum found by ICM tend to deteriorate as the signal becomes more degraded.

The fact that ICM cannot allow increases in energy at any stage can lead to a situation where the process easily becomes trapped in an "avoidable" local minima. For example, suppose that our MAP scene consists of a small block of

foreground pixels, in an otherwise uniform background. If the initial estimate consists entirely of background pixels, then this scene will already be a low energy configuration since it incurs no smoothness penalty. Pixel-wise correction of the block would not occur using ICM if the additional smoothness penalty incurred by revising the first pixel, outweighed the associated reduction in the data fidelity penalty. If a few pixels of the block could be corrected, then each additional connected pixel would make less of an impact on the smoothness penalty. This particular problem might be overcome by allowing groups of pixels to be updated simultaneously. Unfortunately to update n pixels simultaneously requires a search over N^n possible colourings, in place of the n searches over N possibilities needed for a sequential updating of the pixels. Unless both n and N are small, the computational feasibility of such a search is questionable.

4.1.3 Problems with simulated annealing

Simulated annealing is a computationally intensive process which can produce far lower energy reconstructions than ICM, albeit requiring more computational effort. The algorithm can allow pixel changes which result in an increase in energy; theoretically, the process will converge to the MAP estimate as the number of sweeps $\rightarrow \infty$ (see Section 2.3.1). Unfortunately, we can neither find, nor implement, the temperature schedules which are necessary and sufficient for this desired asymptotic result. The problems with this procedure are of a different nature from those described above for ICM. Some of these problems relate to the computational aspects of annealing when there are only a finite number of sweeps available; the others concern how well the sampling mechanisms perform at each particular temperature.

First, it is not clear how well the algorithm can perform when it is only allowed a finite number of sweeps. Single-site update procedures only permit a slow propagation of "information" through the image, and may have a correspondingly slow rate of convergence. Processing on various spatial scales might help to accelerate this convergence. Two multiple-site update algorithms which attempt to do this are the cascade algorithm, and the renormalisation group approach, described in Sections 4.2 and 4.3 respectively.

A second problem for simulated annealing is one which afflicts sampling algorithms of the single-site Hastings type when the target distribution exhibits multi-modality. In the fixed temperature case, successive realisations generated by the sampler can become increasingly dependent as a local minimum is approached. Unless it is possible to escape from the region of attraction by a series of single pixel updates, each with reasonable probability, then the process may spend a long time restricted to this region. The phenomenon is known as critical slowing-down; for a more thorough discussion see Besag & Green (1992), or Sokal (1989). In the

annealing case, this problem is more likely to happen at low temperatures where the chance of any increase in energy is small. In theory, simultaneous updating of pixels appears to be a solution to this problem. In practise, it is not clear how to formulate the corresponding proposal distribution of the Hastings algorithm in such a way that sampling from this $q(\,)$ remains tractable. In Section 2.2.3, this difficulty was stated in introducing the Metropolis algorithm and Gibbs sampler. One workable set of proposals is suggested by the multigrid approach, described in Section 4.5. Alternatively, a modification involving auxiliary variables can be employed, such as is done by the Swendsen and Wang algorithm, discussed in Section 4.4. Either of these two approaches could be incorporated into the annealing process.

4.2 The cascade algorithm

The cascade algorithm is proposed by Jubb & Jennison (1991) as a simple and efficient way of adapting the ICM algorithm to very noisy data. They are motivated by the task of restoring an unblurred image with low signal-to-noise ratio. To effectively increase the signal-to-noise ratio, they average records over non-overlapping 2×2 blocks of pixels. The result could be thought of as the record for an image consisting of fewer, "big" pixels.

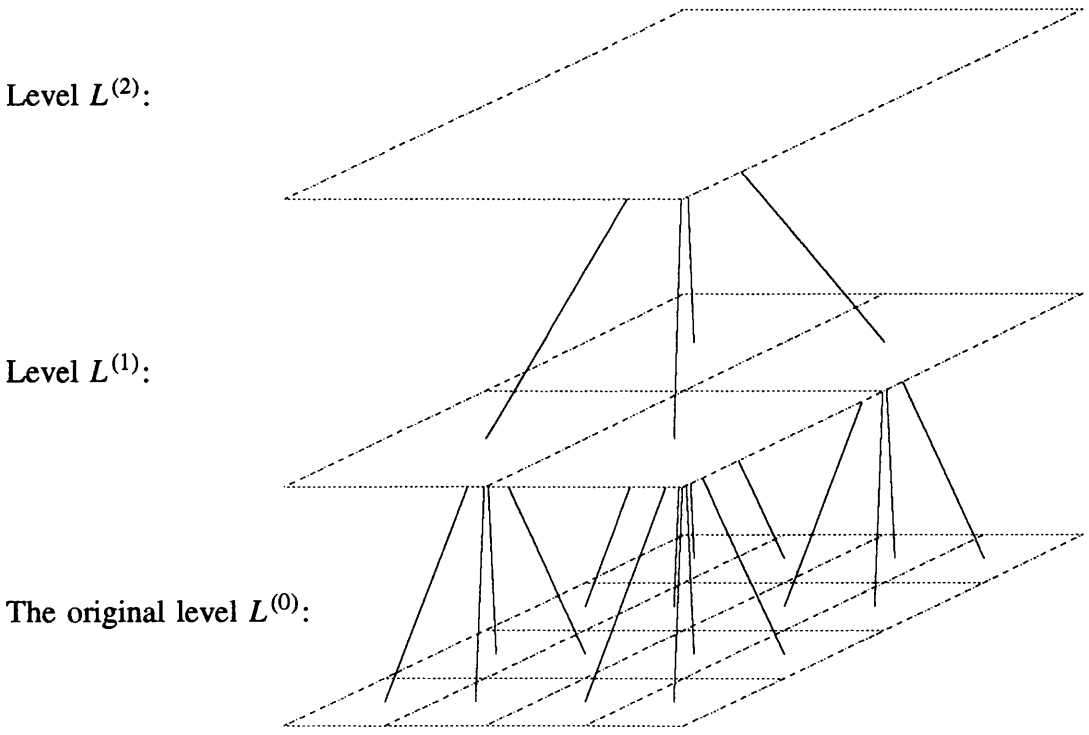


Figure 4.1 Different levels of the pixel grids after 2×2 blocking.

The aggregation and averaging step could be repeated to produce successively coarser resolution images, until eventually the image consists of just a single, large pixel (with slight modifications when the pixel grid is not of suitable dimensions). This idea is illustrated in Figure 4.1, where the original level of pixel grid is denoted by $L^{(0)}$, the 2×2 blocked grid by $L^{(1)}$, and so on. With each averaging of the records, comes a reduction in variance; if the noise variance for level $L^{(0)}$ pixels is σ^2, then the variance of the records at level $L^{(1)}$ is $\sigma^2/4$.

Jubb and Jennison process the cascade of images in a coarse-to-fine sequence, starting with the single pixel image. In the example in their paper, they work with a binary scene using an Ising model (Equation (3.3)). Their energy function on $L^{(0)}$ is of the form

$$H(\mathbf{X}) = \sum_{<s,t>} \beta\, I_{[\mathbf{X}_s \neq \mathbf{X}_t]} + (2\sigma^2)^{-1} \sum_{s \in L^{(0)}} (\mathbf{Y}_s - \mathbf{X}_s)^2,$$

where the notation $<s,t>$ is introduced to denote the clique-pair pixels s and t, and $I_{[\mathbf{X}_s \neq \mathbf{X}_t]}$ is the indicator function taking the value 1 when $\mathbf{X}_s \neq \mathbf{X}_t$, and 0 otherwise. They choose to use the same form of the Markov random field model at all grid levels; the parameter β which represents the attractive strength between neighbouring pixels is held constant for all size pixels. If on the m^{th} level pixel grid $L^{(m)}$, we denote the pixel values by $\mathbf{X}^{(m)}$, and the aggregated records by $\mathbf{Y}^{(m)}$, then the energy function used by Jubb and Jennison at level m is

$$H(\mathbf{X}^{(m)}) = \sum_{<s,t>} \beta\, I_{[\mathbf{X}_s^{(m)} \neq \mathbf{X}_t^{(m)}]} + (2\frac{\sigma^2}{4^m})^{-1} \sum_{s \in L^{(m)}} (\mathbf{Y}_s^{(m)} - \mathbf{X}_s^{(m)})^2,$$

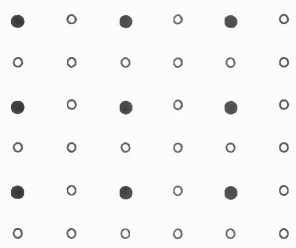
where the cliques $<s,t>$ are now adjacent pixels in $L^{(m)}$. ICM is used at all levels to attempt to minimise the appropriate level energy function. Since the higher levels have fewer pixels, less work is required at these levels. The restoration at any level is passed as the starting point for the next finer level, by initialising each group of four finer pixels to take the value of the corresponding coarser pixel.

Cascade in conjunction with ICM is certainly computationally cheap; it is also simple to understand and implement. In the noisy example given in the paper, the cascade implementation of ICM does seem to be an improvement on the usual single-site update version. However, it is possible to raise objections to certain aspects of the algorithm, in particular the definition of the sequence of energy functions, and the extensions to blurred images. In Chapter 5, we will return to the cascade algorithm, and propose various modifications which will address these objections. We will also demonstrate an extended implementation which appears to improve the algorithm's performance in conjunction with ICM. The behaviour of this modified algorithm, particularly in conjunction with simulated annealing, will be investigated in Chapter 6.

4.3 The renormalisation group approach

The Renormalised Group Approach (RGA) was proposed by Gidas in his 1989 paper; again a sequence of increasingly coarse pixel grids is involved. The rationale given for the range of resolution levels is as an intuitive interpretation of the relation between the viewing distance and the scale of features which can be perceived. Unlike cascade, the links between different levels in this algorithm are determined theoretically so that, under certain conditions, there is guaranteed convergence to the MAP estimate of the image at the finest level. Gidas describes the RGA as an algorithmic framework within which problem dependent choices can be made. In an attempt to outline this complex algorithm, we shall use one specific choice for the grid coarsening procedure and the related probabilities linking consecutive levels. In Gidas' paper, this is an example of the simpler RGA2 algorithm.

The algorithm has two stages, renormalisation which acts in a fine-to-coarse grid direction, and processing which reverses this in acting coarse-to-fine. The renormalisation step generates a sequence of coarser and coarser grids, again denoted $L^{(0)} \rightarrow L^{(1)} \rightarrow \ldots \rightarrow L^{(M)}$ ($L^{(0)}$ is the original full grid), although these are not necessarily formed by 2×2 blocking as they were with cascade. The number of levels, M, and the formation of the grids $L^{(0)}, \ldots, L^{(M)}$ can be selected for a particular problem either for computational convenience, or to take into account some relevant prior information. Gidas gives several examples of grid-coarsening procedures; the example we shall consider here is the so-called "coarsening by decimation". In this case, in passing from one grid to the next coarsest, we retain only every other pixel in both grid directions. This scheme is depicted below, with the decimated pixels of $L^{(m-1)}$ denoted by ∘ and the pixels which remain to form $L^{(m)}$ denoted by •.



It will be convenient to denote a realisation of the image $\mathbf{X}^{(m)}$, on grid $L^{(m)}$, by $\mathbf{x}^{(m)}$. From modelling considerations, we will already have an energy $H(\mathbf{X}^{(0)})$ defined on the original image grid $L^{(0)}$. We now require an associated energy for each of the coarser lattices $L^{(m)}$. Gidas proposes defining the energies iteratively from $H(\mathbf{X}^{(0)})$ at a positive temperature τ. He introduces the conditional probability, $P^{(m)}(\mathbf{X}^{(m)}=\mathbf{x}^{(m)} | \mathbf{X}^{(m-1)}=\mathbf{x}^{(m-1)})$, of a particular image $\mathbf{x}^{(m)}$ on the grid $L^{(m)}$, given the image $\mathbf{x}^{(m-1)}$ on the one-step finer grid $L^{(m-1)}$. The

specification of τ, and these conditional probabilities $P^{(m)}(\mathbf{x}^{(m)}|\mathbf{x}^{(m-1)})$, are the two remaining problem dependent choices. The sequence of coarse grid energies (in Gidas' terminology, Hamiltonians), $H_\tau^{(0)} \rightarrow H_\tau^{(1)} \rightarrow \ldots \rightarrow H_\tau^{(M)}$ is then generated using the standard probability relationship,

$$P(\mathbf{X}^{(m)}=\mathbf{x}^{(m)}) = \sum_{\mathbf{x}^{(m-1)}} P^{(m)}(\mathbf{x}^{(m)}|\mathbf{x}^{(m-1)})\, P(\mathbf{X}^{(m-1)}=\mathbf{x}^{(m-1)}), \quad m=1,\ldots, M.$$

$$\text{So} \quad \exp(-H_\tau^{(m)}(\mathbf{x}^{(m)})) = \sum_{\mathbf{x}^{(m-1)}} P^{(m)}(\mathbf{x}^{(m)}|\mathbf{x}^{(m-1)}) \exp(-H_\tau^{(m-1)}(\mathbf{x}^{(m-1)})) \tag{4.1}$$

$$\text{where} \qquad H_\tau^{(0)}(\mathbf{x}^{(0)}) = \frac{H(\mathbf{x}^{(0)})}{\tau}. \tag{4.2}$$

It is Equation (4.1) which could be expressed as $H_\tau^{(m)}=R^{(m)}(H_\tau^{(m-1)})$, where $R^{(m)}$ is the "renormalisation group transform" giving its name to the algorithm. Gidas states that in general the calculation of $H_\tau^{(m)}$, $1 \le m \le M$, can only be done approximately, and a great deal of the paper is devoted to tackling this implementation problem.

Gidas suggests that a natural choice of conditional probability for our coarsening by decimation example, is

$$P^{(m)}(\mathbf{x}^{(m)}|\mathbf{x}^{(m-1)}) = \prod_{s \in L^{(m)}} \delta_{\mathbf{x}_{(s)}^{(m)} \mathbf{x}_s^{(m-1)}} \tag{4.3}$$

where $\delta_{\mathbf{x}_{(s)}^{(m)} \mathbf{x}_s^{(m-1)}}$ takes the value 1 when $\mathbf{x}_{(s)}^{(m)}=\mathbf{x}_s^{(m-1)}$, and 0 otherwise (the notation $\mathbf{x}_{(s)}^{(m)}$ is taken to indicate the value of the pixel in $L^{(m)}$ generated by non-decimation of pixel s in $L^{(m-1)}$). So in our example, given an $\mathbf{x}^{(m-1)}$ there is only one $\mathbf{x}^{(m)}$ with positive probability, and the energy functions at consecutive levels are related according to $\exp(-H_\tau^{(m)}(\mathbf{x}^{(m)})) = \sum_{\substack{\mathbf{x}^{(m-1)} \\ \mathbf{x}^{(m)} \text{ on } L^{(m)}}} \exp(-H_\tau^{(m-1)}(\mathbf{x}^{(m-1)}))$.

A fine-to-coarse scheme has now been outlined for generating different resolution image grids, and their corresponding linked energy functions. Next we need to consider the processing step, the overall aim of which is to find the minimiser of $H(\mathbf{X}^{(0)})$. Since each $L^{(m)}$ contains fewer pixels than $L^{(m-1)}$, the processing begins with the smallest problem, minimising $H_\tau^{(M)}(\mathbf{X}^{(M)})$ on the coarsest grid $L^{(M)}$. The minimising technique proposed is simulated annealing, and Gidas makes the assumption that we can find the minimiser, $\tilde{\mathbf{x}}^{(M)}$. Having located $\tilde{\mathbf{x}}^{(M)}$, the processing moves onto the next finer grid $L^{(M-1)}$. Now, rather than searching on the space of all possible $\mathbf{x}^{(M-1)}$, the algorithm only considers those $\mathbf{x}^{(M-1)}$ which satisfy the constraint that the conditional probability of $\mathbf{x}^{(M)}$ given $\mathbf{x}^{(M-1)}$ is maximised by $\tilde{\mathbf{x}}^{(M)}$. This condition reduces the computational complexity of the search. For our particular coarsening example, this condition says that we will only consider $\mathbf{x}^{(M-1)}$ which maximise $P(\tilde{\mathbf{x}}^{(M)}|\mathbf{x}^{(M-1)})$. From Equation (4.3), this conditional probability is 1 if $\mathbf{x}^{(M-1)}$ and $\tilde{\mathbf{x}}^{(M)}$ match on $L^{(M)}$,

and zero otherwise. So, in effect, we have every second pixel in both directions on $L^{(M-1)}$ fixed, and the minimisation is attempting to fill in the "gaps" created by coarsening. This constrained minimum is denoted $\tilde{\mathbf{x}}^{(M-1)}$.

Continuing up the grid resolutions, if $\tilde{\mathbf{x}}^{(m)}$ is the constrained minimum of $H_\tau^{(m)}$, then in minimising $H_\tau^{(m-1)}$ we only consider those $\mathbf{x}^{(m-1)}$ satisfying

$$P^{(m)}(\tilde{\mathbf{x}}^{(m)}|\mathbf{x}^{(m-1)}) = \max_{\mathbf{x}^{(m)}} P^{(m)}(\mathbf{x}^{(m)}|\mathbf{x}^{(m-1)}). \qquad (4.4)$$

In our example, at each increased resolution grid we are optimising over the values of the three-quarters of the pixels reintroduced at that level. For an N colour scene, viewed at a resolution resulting in n pixels, this constraining of the search space reduces its dimension from N^n to $N^{3n/4}$.

Gidas proves that for τ below a certain critical value $\bar{\tau}$, an RGA2 algorithm, as outlined above, will converge to the global minimiser of H(**X**). The constant $\bar{\tau}$ will depend on the energy function H(**X**); there are connections with the dependence on H(**X**) of the theoretically correct annealing schedule. There is a similar, although more complicated, result for the more general RGA algorithm.

In theory, the renormalised group approach provides an appealing way of generating connected multiresolution image grids whose processing stages reduce computational effort, and produce the MAP estimate under certain constraints. In practise, the coarse energy functions will be difficult to compute, and approximations may be needed. This is also true of the bounds on the constants necessary for convergence. The optimisation at each level will still be subject to the usual problems (ICM is too inaccurate, simulated annealing is too slow), albeit on a reduced scale as the search space is smaller. These implementation problems, particularly the calculation of the coarse grid energies, are all tackled in the paper.

4.4 Swendsen and Wang's algorithm

4.4.1 Original proposal

Swendsen & Wang (1987) propose a new sampling technique which uses multiple-site updates in an attempt to avoid critical slowing-down. This section deals with the algorithm as they describe it, solely for simulating from N-colour Ising models observed without noise, and at a fixed temperature.

The algorithm begins with an arbitrary configuration on the grid; the aim is to generate samples from the Ising model with parameter β. Using the notation $<s,t>$ to denote pairwise pixel cliques, this model has a probability density function which can be written in the form

$$p(\mathbf{X}=\mathbf{x}) = \exp\Big(-\sum_{<s,t>} \beta\, I_{[\mathbf{x}_s \neq \mathbf{x}_t]}\Big) / Z, \qquad \mathbf{x}_s \in \{0, 1, \ldots, N-1\},\ \forall s \in S^0. \quad (4.5)$$

The algorithm begins by introducing an additional set of bond variables, one for each clique interaction, and each taking the value "on" or "off". Given the current realisation of the pixel values, we generate a realisation of these bond variables. If pixels s and t in the clique $<s,t>$ take different values, then the bond between them is set to "off". If the pixels take the same value, the algorithm sets the bond to "on" with probability $1-e^{-\beta}$, otherwise the bond is set to "off" . The bond variables are conditionally independent given the current scene. Neighbouring pixels taking different values do not have bonds connecting them, those taking the same value may or may not be bonded. Clusters are then defined to be either groups of pixels connected by "on" bonds, or singleton pixels surrounded by "off" bonds. This partitions the scene into disjoint clusters, each of which is composed of pixels taking a single colour. Now, given the current values of the bond variables, we can generate a new realisation for the pixel variables. The current pixel colourings are discarded, and a random colouring is selected independently for each bond-defined cluster; this gives the next realisation. The values of the bond variables are then discarded; this completes one cycle of the algorithm. The procedure can then be repeated from the bond formation stage.

Clearly there is the potential for large numbers of pixels to change from one realisation to the next. The algorithm does appear to be effective in avoiding critical slowing-down. Swendsen and Wang only define and justify the algorithm in the rather limited case of probability density functions of the form of Equation (4.5). The proof relies upon showing that each cycle of the algorithm satisfies detailed balance, Equation (2.12), with respect to the Ising model in which we are interested. We will leave the details to the next section, where the extensions proposed for sampling from more general density functions, including the Ising model as a special case, are discussed.

4.4.2 Extensions

A generalisation of the standard Swendsen and Wang algorithm is suggested by Edwards & Sokal (1988). A more comprehensive discussion of this generalisation, and suggestions for its implementation, are given by Green (1991); we shall follow the latter.

Suppose we are attempting to simulate from a density function defined on the usual sample space $N^{|S^0|}$, and which can be written in the form

$$p(\mathbf{X}=\mathbf{x}) = \frac{1}{Z} \prod_{s:s\in S^0} a_s(\mathbf{x}_s) \prod_{<s,t>} b_{s,t}(|\mathbf{x}_s-\mathbf{x}_t|), \tag{4.6}$$

where $\{a_s(\,)\}$ and $\{b_{s,t}(\,)\}$ are finite, strictly positive functions with each $b_{s,t}(\,)$ non-increasing. For the Ising case discussed in the last section, the $\{a_s(\,)\}$ are identically 1, while $b_{s,t}(|\mathbf{x}_s-\mathbf{x}_t|)=\exp(-\beta \mathrm{I}_{[\mathbf{x}_s\neq\mathbf{x}_t]})$.

The first step in the construction is to introduce a vector **E** of additional bond variables, one for each pairwise clique interaction, $\mathbf{E}=\{\mathbf{E}_{<s,t>}\}$. A realisation of any particular $\mathbf{E}_{<s,t>}$, denoted $\mathbf{e}_{<s,t>}$, can take the same range of values as the components of **X**, namely $\{0, 1, \ldots, N-1\}$. By a suitable choice of the conditional distribution of **E** given the scene **x**, we can define the $\mathbf{E}_{<s,t>}$ to be conditionally independent given **x**. The intention is to construct the **e** in such a way as to induce a particular cluster system on the set of pixels. Then given **e**, the $\mathbf{X}_s$ in different clusters can be shown to be conditionally independent. Such a system would permit a sequential update of the **E** given **x**, followed by a cluster update of the **X** given **e**. The **X** follows the distribution given in Equation (4.6) throughout. Before giving Green's proof of the validity of the algorithm for sampling from this $p(\mathbf{x})$, we will describe the construction of **e** and $p(\mathbf{e}|\mathbf{x})$.

Let $\quad c_{s,t}(e) = \begin{cases} b_{s,t}(e)-b_{s,t}(e+1), & \text{if } e=0, 1, \ldots, N-2 \\ b_{s,t}(N-1), & \text{if } e=N-1, \end{cases}$

thus $\quad b_{s,t}(d) = c_{s,t}(d) + c_{s,t}(d+1) + \ldots + c_{s,t}(N-1), \qquad d=0, 1, \ldots, N-1$

$$= \sum_{e=0}^{N-1} c_{s,t}(e)\, \mathrm{I}_{[e\geq d]}.$$

Using the above, the $\{\mathbf{E}_{<s,t>}\}$ are defined to be conditionally independent given **x**

$$p(\mathbf{e}_{<s,t>}|\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{e}_{<s,t>}<|\mathbf{x}_s-\mathbf{x}_t| \\ \dfrac{c_{s,t}(\mathbf{e}_{<s,t>})}{b_{s,t}(|\mathbf{x}_s-\mathbf{x}_t|)}, & \text{if } \mathbf{e}_{<s,t>}\geq|\mathbf{x}_s-\mathbf{x}_t|. \end{cases} \tag{4.7}$$

To see that that is a proper distribution, notice that each term is non-negative (using the positive, non-increasing properties of $b_{s,t}(\,)$ in the definition of $c_{s,t}(\,)$), and $\sum_{\mathbf{e}_{<s,t>}=0}^{N-1} p(\mathbf{e}_{<s,t>}|\mathbf{x}) = \dfrac{1}{b_{s,t}(|\mathbf{x}_s-\mathbf{x}_t|)} \sum_{\mathbf{e}_{<s,t>}=|\mathbf{x}_s-\mathbf{x}_t|}^{N-1} c_{s,t}(\mathbf{e}_{<s,t>}) = 1.$

Equation (4.7) defines the sampling distribution from which a realisation of **E** given **x** must be drawn. The distribution for the new realisation for the scene given the current values of the bond variables is $p(\mathbf{x}|\mathbf{e})$. To obtain this conditional distribution, we will work via the joint distribution,

$$p(\mathbf{x},\mathbf{e}) = p(\mathbf{e}|\mathbf{x})\, p(\mathbf{x})$$

$$= \prod_{<s,t>} \frac{c_{s,t}(\mathbf{e}_{<s,t>})}{b_{s,t}(|\mathbf{x}_s-\mathbf{x}_t|)}\, \mathrm{I}_{[|\mathbf{x}_s-\mathbf{x}_t|\leq\mathbf{e}_{<s,t>}]}\, \frac{1}{Z} \prod_{s:s\in S^0} a_s(\mathbf{x}_s) \prod_{<s,t>} b_{s,t}(|\mathbf{x}_s-\mathbf{x}_t|)$$

$$= \frac{1}{Z} \prod_{s:s\in S^0} a_s(\mathbf{x}_s) \prod_{<s,t>} c_{s,t}(\mathbf{e}_{<s,t>})\, \mathrm{I}_{[|\mathbf{x}_s-\mathbf{x}_t|\leq\mathbf{e}_{<s,t>}]}. \tag{4.8}$$

So $\quad p(\mathbf{x}|\mathbf{e}) \propto \prod_{s:s\in S^0} a_s(\mathbf{x}_s) \prod_{<s,t>} \mathrm{I}_{[|\mathbf{x}_s-\mathbf{x}_t|\leq\mathbf{e}_{<s,t>}]}. \tag{4.9}$

Equation (4.9) shows that the conditional distribution of $\mathbf{X}$ given $\mathbf{e}$ is proportional to $\prod_{s:s\in S^0} a_s(\mathbf{x}_s)$, but only for those $\{\mathbf{x}\}$ which satisfy the condition $|\mathbf{x}_s-\mathbf{x}_t|\le\mathbf{e}_{<s,t>}$, $\forall$ clique pairs $<s,t>$. It is clear that if $\mathbf{e}_{<s,t>}=N-1$, then $\mathbf{X}_s$ and $\mathbf{X}_t$ are free to take any two values in $\{0, 1, \ldots, N-1\}$, since their absolute difference will always satisfy this condition. We will therefore define clusters to be the equivalence classes generated by grouping together pixels between which some path exists with $\mathbf{e}_{<s,t>}<N-1$ for all the cliques in the path. Then given $\mathbf{e}$, the $\{\mathbf{X}_s\}$ in different clusters are conditionally independent. Notice that clusters in the extended Swendsen and Wang algorithm are no longer restricted to taking a single colour. However, they can still be updated independently.

To illustrate this construction, and to confirm that the Ising case in Section 4.4.1 is a particular example, consider the situation where the $a_s(\,)$ are identically 1 and $b_{s,t}(|\mathbf{x}_s-\mathbf{x}_t|)=\exp(-\beta I_{[\mathbf{x}_s\neq\mathbf{x}_t]})$. Here

$$c_{s,t}(e) = \begin{cases} 1-\exp(-\beta), & \text{if } e=0 \\ \exp(-\beta), & \text{if } e=N-1 \\ 0, & \text{if } e=1, \ldots, N-2, \end{cases}$$

$$\text{and} \quad b_{s,t}(d) = \begin{cases} 1, & \text{if } d=0 \\ \exp(-\beta), & \text{if } d=1, \ldots, N-1\ . \end{cases}$$

The conditional probability of a bond $\mathbf{E}_{<s,t>}$ given the current scene $\mathbf{x}$ is then

$$p(\mathbf{e}_{<s,t>}|\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{e}_{<s,t>}<|\mathbf{x}_s-\mathbf{x}_t| \\ \dfrac{c_{s,t}(\mathbf{e}_{<s,t>})}{b_{s,t}(|\mathbf{x}_s-\mathbf{x}_t|)}, & \text{if } \mathbf{e}_{<s,t>}\ge|\mathbf{x}_s-\mathbf{x}_t| \end{cases}$$

$$= \begin{cases} 1-\exp(-\beta), & \text{if } \mathbf{e}_{<s,t>}=0, \text{ and } |\mathbf{x}_s-\mathbf{x}_t|=0 \\ \exp(-\beta), & \text{if } \mathbf{e}_{<s,t>}=N-1, \text{ and } |\mathbf{x}_s-\mathbf{x}_t|=0 \\ 1, & \text{if } \mathbf{e}_{<s,t>}=N-1, \text{ and } |\mathbf{x}_s-\mathbf{x}_t|\neq 0 \\ 0, & \text{otherwise.} \end{cases}$$

So the bond variables take only two values, 0 and $N-1$, and these could be identified as "on" and "off" respectively. A bond is always "off" if it is associated with a neighbourhood interaction where $\mathbf{x}_s\neq\mathbf{x}_t$. If $\mathbf{x}_s=\mathbf{x}_t$, then with probability $1-\exp(-\beta)$, $\mathbf{e}_{<s,t>}=0$, otherwise the bond is "off". Since clusters are defined as pixels linked by bonds satisfying $\mathbf{e}_{<s,t>}<N-1$, clusters will take a single colour. For updating, $p(\mathbf{x}|\mathbf{e})\propto\prod_{<s,t>} I_{[|\mathbf{x}_s-\mathbf{x}_t|\le\mathbf{e}_{<s,t>}]}$. So clusters are required to remain as a single colour, but this single value for each cluster can be chosen at random from the N possibilities. This is precisely the original Swendsen and Wang algorithm.

We can now state the proof, given by Green (1991), that this generalised Swendsen and Wang algorithm is valid for producing samples from $p(\mathbf{x})$. Since we have just seen that the original algorithm is a particular example of this generalisation, it is covered by this argument.

In Section 2.2.2, we stated the conditions under which an iteratively generated Markov chain converged in distribution to the equilibrium distribution, in this case given by Equation (4.6), from any arbitrary starting configuration. These conditions were that the Markov chain transition function $P(\mathbf{x},\mathbf{x}')$ was irreducible, aperiodic, and satisfied detailed balance with respect to the equilibrium distribution, $P(\mathbf{x},\mathbf{x}')p(\mathbf{x})=P(\mathbf{x}',\mathbf{x})p(\mathbf{x}')$, $\forall\ \mathbf{x},\ \mathbf{x}'\in\Omega$.

In this algorithm, the transitions from $\mathbf{x}$ to $\mathbf{x}'$ are two-step; given some $\mathbf{x}$, we generate a realisation $\mathbf{e}$, then given this $\mathbf{e}$, we generate a new $\mathbf{x}'$. To find the probability of generating $\mathbf{x}'$ from $\mathbf{x}$, we need to consider the sum of probabilities via all possible configurations of the bond variables,

$$P(\mathbf{x},\mathbf{x}') = \sum_{\mathbf{e}} p(\mathbf{e}|\mathbf{x})\ p(\mathbf{x}'\,|\mathbf{e})$$

$$= \sum_{\mathbf{e}} \frac{p(\mathbf{x},\mathbf{e})}{p(\mathbf{x})}\ \frac{p(\mathbf{x}',\mathbf{e})}{p(\mathbf{e})}.$$

$$\Rightarrow \qquad P(\mathbf{x},\mathbf{x}')\ p(\mathbf{x}) = \sum_{\mathbf{e}} \frac{p(\mathbf{x},\mathbf{e})\ p(\mathbf{x}',\mathbf{e})}{p(\mathbf{e})}$$

and is symmetric in $\mathbf{x}$ and $\mathbf{x}'$, and so detailed balance will be satisfied.

To demonstrate irreducibility and aperiodicity, we will consider the special realisation $\mathbf{e}^*$ of the bond variables satisfying $\mathbf{e}^*_{<s,t>}=N-1$, $\forall <s,\ t>$. By the definition of clusters as groups of pixels connected by $\mathbf{e}_{<s,t>}<N-1$, $\mathbf{e}^*$ induces $|S^0|$ clusters each consisting of a single pixel. It also implies that the condition $|\mathbf{x}_s-\mathbf{x}_t|\leq\mathbf{e}^*_{<s,t>}$ is satisfied for all clique pairs $<s,t>$. From the definition of the conditional distribution in Equation (4.7), and the positivity of $b_{s,t}(\ )$,

$$p(\mathbf{e}^*_{<s,t>}|\mathbf{x}) = \frac{b_{s,t}(N-1)}{b_{s,t}(|\mathbf{x}_s-\mathbf{x}_t|)} > 0, \qquad \forall\ <s,t>,\ \forall\mathbf{x}\in\Omega.$$

That is, it is always possible to realise $\mathbf{e}^*$ from any $\mathbf{x}$. Then, since under $\mathbf{e}^*$ the constraints on $|\mathbf{x}'_s-\mathbf{x}'_t|$ are never violated by any $\mathbf{x}'$, from Equation (4.9)

$$p(\mathbf{x}'\,|\mathbf{e}^*) \propto \prod_{s\in S^0} a_s(\mathbf{x}'_s), \qquad \forall\mathbf{x}'\in\Omega.$$

Since the $a_s(\ )$ are strictly positive, $p(\mathbf{x}'\,|\mathbf{e}^*)>0$, $\forall\ \mathbf{x}'\in\Omega$. So it is possible to generate any $\mathbf{x}'$ given the realisation $\mathbf{e}^*$. Therefore $P(\mathbf{x},\mathbf{x}')>0$, $\forall\ \mathbf{x},\ \mathbf{x}'\in\Omega$. This demonstrates irreducibility. Also, since $P(\mathbf{x},\mathbf{x})>0$, $\forall\mathbf{x}\in\Omega$, the aperiodicity requirement is satisfied.

Green points out that this extended algorithm can also be generalised further to sample from posterior densities, provided that the likelihood $p(\mathbf{y}|\mathbf{x})$ is of the pixel-wise independent form $p(\mathbf{y}|\mathbf{x})=\prod_{s\in S^0} p(\mathbf{y}_s|\mathbf{x}_s)$. In this situation, the posterior is proportional to the product of this likelihood with the prior given in Equation (4.6). The extension is possible because, throughout the analysis, we can replace $\prod_{s\in S^0} a_s(\mathbf{x}_s)$ by $\prod_{s\in S^0} a_s(\mathbf{x}_s)p(\mathbf{y}_s|\mathbf{x}_s)$.

This extended Swendsen and Wang algorithm satisfies all the conditions necessary for the distribution of the generated $\{\mathbf{x}\}$ to converge to the desired target distribution. By introducing the auxiliary bond variables $\mathbf{E}$ which "kill" interactions between some components of $\mathbf{X}$, it provides a viable method of updating groups of pixels while maintaining detailed balance. The major implementation problem which remains is how to sample efficiently from $p(\mathbf{x}|\mathbf{e})$, Equation (4.9), given the awkward restrictions on the sample space. Green suggests various possible approaches, but these are speculative.

It is also worth noting that there are two apparent restrictions on the application of this algorithm. First, the prior, Equation (4.6), permits only pairwise difference clique interactions. This would appear to rule out second and third order models of the types described in Section 3.3, since in these models cliques involve at least three pixels.

Secondly, the extension to posterior densities is only applicable when the record does not involve any blurring. One possible approach to including blurring would be to decompose the blurred likelihood into terms involving either single components of $\mathbf{X}$, or paired difference terms. Using the fact that the sum of the blurring coefficients γ_{s-t} equals 1, algebraic manipulation gives the expansion,

$$\lambda \parallel \mathbf{y}-\mathbf{K}\mathbf{x} \parallel^2 = \lambda \sum_{s\in S} \{ \mathbf{y}_s^2 - 2\mathbf{y}_s \sum_{t\in B_s} \gamma_{s-t}\mathbf{x}_t + \sum_{t\in B_s} \gamma_{s-t}\mathbf{x}_t^2 + \sum_{t\in B_s} \sum_{r\in B_s, r\neq t} \gamma_{s-t}\gamma_{s-r}\mathbf{x}_r^2 - \sum_{t\in B_s} \sum_{r\in B_s, r\neq t} \gamma_{s-t}\gamma_{s-r}(|\mathbf{x}_t-\mathbf{x}_r|)^2 \}. \quad (4.10)$$

This expansion would suggest suitably reordering the first four summation terms with a view to including the exponential of their negative values with the $\prod_{s\in S^0} a_s(\mathbf{x}_s)$ term. The remaining expansion terms could then be incorporated into $\prod_{<s,t>} b_{s,t}(|\mathbf{x}_s-\mathbf{x}_t|)$. Unfortunately, we require that each pixel difference function is non-increasing in $|\mathbf{x}_s-\mathbf{x}_t|$. Certain prior $b_{s,t}(\,)$ combined with the relevant terms of $\exp(\lambda \sum_{s\in S^0} \sum_{t\in B_s} \sum_{r\in B_s, r\neq t} \gamma_{s-t}\gamma_{s-r}(|\mathbf{x}_t-\mathbf{x}_r|)^2)$ may not satisfy this condition. We have not considered any further approaches, however it seems likely that any possible inclusion of blurring would increase the complexity of the algorithm.

4.5 Multigrid techniques

4.5.1 General numerical analysis setting

Multigrid techniques were introduced in an attempt to accelerate the convergence of iterative numerical methods for solving partial differential equations. A description of the background, theory and applications in this field is given by Briggs (1987). The primary motivation comes from the observation that while iterative methods are efficient at eliminating high frequency components of the error, they cope less well with low frequency components. After a certain number of iterations, the high frequency components of the error will have been minimised, and the residual errors will tend to be smoothly varying. Convergence then slows as a site-by-site iterative procedure finds it difficult to overcome this slow variation. By considering the residual errors on a coarser resolution grid, they appear to be less smooth, and to have higher frequency. So, the same iterative methods, known to be effective at high frequency, could be applied on the coarse grid to approximately solve the residual equation. This approximation could then be transferred back to the fine grid, using the ideas of residual correction to improve the initial solution.

So, in outline, the multigrid approach involves an iterative method for solving a system on a given grid, a restriction scheme for generating the residual system on a specified coarser grid, and an interpolation scheme for expanding the residual corrections back to the finer grid. There will also be problem dependent choices of a grid coarsening scheme, and the number of grid levels. Processing begins with the finest grid. A residual correction can be made on the next coarsest grid. However before passing this correction down to the full grid, it could itself be refined by working on a still coarser grid. There are various forms of these grid cycles, each built from moves between consecutive levels in the coarseness scale. The final processing will be on the finest grid. It is hoped that the multigrid approach will help to eliminate components of the error at all frequencies.

4.5.2 Applications to imaging

In the imaging context, the problem to be addressed does not fit exactly into the framework described above. We are attempting either to sample, or to minimise, an energy function on a grid defined initially by the data-collection. This is rather different than solving a differential equation approximated onto a grid of arbitrary resolution. There are "image analysis" exceptions to this, see for example Terzopoulos (1986) which uses standard multigrid techniques to solve the differential equations of "shape-from-shading", optical flow, and other problems. It is still possible however, in the type of problem we are considering, to use the general multigrid concepts of corrections on varying resolution grids.

Sokal (1989) discusses a multigrid scheme for simulating from some density defined for variables on a grid. We begin with some initial image $\mathbf{x}$ on the complete grid, for example by using a standard single-site update algorithm. We employ the grid reduction scheme which associates each node in the one-stage reduced grid with a distinct block of four pixels in the original grid, and so on (a scheme of this type is illustrated in Figure 4.1). Unlike in cascade where the higher levels could be considered as coarser pixel grids, here the variables at higher grid nodes in some way represent "residual error". Denote the variables on the m^{th} level grid, $L^{(m)}$, by $\mathbf{Z}^{(m)}$, and realisations of these variables by $\mathbf{z}^{(m)}$, $m=1, \ldots, M$. At all levels m, each component of the $\mathbf{Z}^{(m)}$ is permitted to take values in the set of integers $\{-(N-1), \ldots, (N-1)\}$. We begin by setting each component of all the $\mathbf{Z}^{(m)}$ to be zero. Notice that we could then express the value of the s^{th} pixel in $L^{(0)}$ as $(\mathbf{x}_s+\mathbf{z}_{(s)}^{(1)}+\ldots+\mathbf{z}_{(s)}^{(M)})$, where $\mathbf{z}_{(s)}^{(m)}$ denotes the realisation of $\mathbf{Z}^{(m)}$ at the node of $L^{(m)}$ which is the unique $L^{(m)}$ "parent" of pixel s of the original grid $L^{(0)}$. When the value of one of the $L^{(m)}$ node variables changes, the value of each of the 4^m associated child pixels in the original grid is shifted additively by the change in that component of $\mathbf{Z}^{(m)}$. The intention is to generate new images on the pixel grid by generating new values for the components of $\mathbf{Z}^{(m)}$. Blocks of pixels are not constrained to take a common value as they were by the cascade algorithm, however the within-block differences in the pixel values are fixed. Irreducibility cannot be satisfied at any level other than the original, since pixels within groups are fixed in their relative values. The block size is dictated by the level of the multigrid; we might move from blocks of side 1, to side 2, to side 4, then back to side 2, and finally back to side 1.

The benefit of this multigrid approach is that the sampling distributions for the components of the $\mathbf{Z}^{(m)}$ may be constructed to be of a computationally simple form, since each component can only take $2N-1$ possible values. A Hastings algorithm, as defined in Section 2.2.2, is used for drawing a sample of the $\mathbf{Z}^{(m)}$ variables. The intention is to maintain detailed balance for the original pixel process, while excluding any values of $\mathbf{Z}^{(m)}$ which would result in any pixel taking a value outside its permitted range, $\{0, \ldots, N-1\}$. Holding all other variables fixed, each $\mathbf{z}^{(m)}$ is associated with a particular pixel image $\mathbf{x}$. Proposing a new value for some component of $\mathbf{Z}^{(m)}$ is equivalent to proposing a new $\tilde{\mathbf{x}}$ which may differ from $\mathbf{x}$ at the 4^m pixels affected by this component. Notice that to propose $\mathbf{x}$ from $\tilde{\mathbf{x}}$, we would need to reverse the change in the value of the reduced grid variable. The standard form for a Hastings algorithm is given in Equation (2.10), involving a proposal distribution, and an acceptance distribution. We can choose some convenient proposal probability distribution $q(\mathbf{z}_s^{(m)}, \tilde{\mathbf{z}}_s^{(m)})$, the probability that the value $\tilde{\mathbf{z}}_s^{(m)}$ is proposed for the s^{th} component of $\mathbf{Z}^{(m)}$, given that it currently

takes the value $\mathbf{z}_s^{(m)}$. Suppose we define the probability of a particular realisation of the $\mathbf{Z}^{(m)}$ to be the probability of the corresponding image $\mathbf{x}$ (suitably renormalised to account for the images which cannot be attained at this level). Then Equation (2.13) gives the acceptance probability for this proposed change to maintain detailed balance for the pixel process as

$$\alpha(\, \mathbf{z}_s^{(m)}, \tilde{\mathbf{z}}_s^{(m)} \,) = \min(\, 1 \,,\, \frac{q(\tilde{\mathbf{z}}_s^{(m)}, \mathbf{z}_s^{(m)})\; p(\tilde{\mathbf{x}})}{q(\mathbf{z}_s^{(m)}, \tilde{\mathbf{z}}_s^{(m)})\; p(\mathbf{x})} \,).$$

This will ensure a zero probability of accepting any value of the $\mathbf{Z}^{(m)}$ which would result in an invalid pixel value in $\tilde{\mathbf{x}}$, since the corresponding $p(\tilde{\mathbf{x}})$ will be zero.

There are also other interpretations of multigrid techniques in imaging. Brandt, Ron & Amit (1985) consider using different shape blocks in a two colour Ising prior. The decision to update is made only after calculating the resulting effect on the finest scale. Kandel, Domany & Brandt (1989) combine the Swendsen and Wang algorithm with a stochastically blocked multigrid approach for Ising models. Here certain pixels are designated as coarse, and reduced grid clusters are formed from these by having bonds "on", "off", or "ignored". The coarse labels are varied to give different levels of the multigrid.

4.6 Some other approaches

4.6.1 Pyramid methods

Image pyramid methods do not fall strictly within the limits of this chapter; they tend to be used for image segmentation, compression, or edge-detection purposes, rather than for processing in the sense which we have been considering. An overview of various properties and applications can be found in Rosenfeld (1984). The methods are briefly mentioned here because of the similarity in their fine-to-coarse-to-fine approach.

The so-called pyramid is a stack of images containing successively fewer pixels. Links between levels are referred to as groups of "child" pixels connected to "parent" pixels. These family relations may be defined deterministically, see for example Cibulskis & Dyer (1984), or stochastically (Montanvert, Meer & Rosenfeld (1991)). We will just outline a simple deterministic case, such as might be used for segmentation.

At every level, beginning at the finest, the child pixels are grouped into regular blocks, each of which is linked to some parent pixel located above it in the next level. The blocks are taken to be overlapping, so that each child has more than one parent. Suppose, for example in comparison to Figure 4.1, that the 2×2 blocking scheme is taken to be overlapping, so that internal pixels are included in four separate blocks. If these blocks are linked either to the pixel directly above in

the pyramid, or to the pixel above with the highest label if this is not uniquely determined, then each internal pixel will have four parents. The value taken by the parent is initialised as the average of its childrens' grey-levels. Depending on the intended use, it would also be possible to pass alternative information up through the pyramid. At this stage, some iteration takes place. Given the parents' current values, each child is examined according to some distance criterion to determine a "best" parent from those available. In the grey-level case, this will be the parent with the grey-level closest to its own. The other parent-child links are then broken, and the parent values recalculated from their remaining children. The relinking (allowing previously broken links to be reformed), and recalculating are repeated until the system stabilises. The next level higher is then considered.

Some pyramid linking algorithms allow child pixels to be orphaned if all the potential parents take sufficiently different values. Merging techniques also exist for joining these orphans to parents further away in the pyramid system if this seems appropriate. When the level of the pyramid designated as the top is reached, we are left with a final set of parents (again some of whom may be merged if this seems appropriate), and any remaining orphans lower in the structure. The segmentation is given by tracing the final values of these nodes back down through the family trees defined by the parent-child links.

4.6.2 Methods in texture analysis

In some applications, the underlying scene may be composed of homogeneous regions formed by some repeated pattern other than a uniform grey-level. These patterns are known as textures. We have not discussed texture analysis; two possible references are Cross & Jain (1983) for a discussion of texture modelling using the Ising model, and Geman & Graffigne (1986) for a discussion of labelling and texture segmentation.

Since textures, by definition, involve the interactions of a number of pixels, texture analysis sometimes involves considering groups of pixels together for both segmentation, and parameter estimation. As an example, Geman, Geman, Graffigne & Dong (1990) consider segmentation and the detection of boundaries between different textures by comparing the histograms of grey-level values in different square pixel blocks. In the segmentation case, the overlapping blocks are chosen "sufficiently large" to capture the characteristic pattern of the texture. The segmentation is generated by comparisons with the four nearest neighbour blocks, and with sixteen randomly chosen blocks to introduce longer range effects. In the boundary detection case, the differences to be considered are between adjacent, non-overlapping pixel blocks.

An approach which has more in common with some of the multiresolution procedures considered in previous sections is given by Bouman & Liu (1991). Their algorithm processes coarse-to-fine, with each four distinct pixels at a finer resolution constituting one coarse pixel at the resolution above (again see Figure 4.1). At each level, the number of distinct textures, and their associated parameters are estimated. Pixel blocks are then assigned to one of these classes by minimising a modified energy function at that level using ICM. In the case where the textures correspond simply to grey-levels, the processing step of their algorithm is the same as in the cascade algorithm discussed in Section 4.2.

Chapter 5: An Extension of Cascade

5.1 The conceptual problems with cascade

5.1.1 Inconsistency of prior models

The cascade algorithm, due to Jubb & Jennison (1991), was described in Section 4.2. It is a multiple-site method, applying ICM to update $2^m \times 2^m$ blocks of pixels with a view to minimising an energy function defined on the m^{th} level grid, $L^{(m)}$. Using the notation of Section 4.2, $\mathbf{X}^{(m)}$ is the image on grid $L^{(m)}$, and $\mathbf{Y}^{(m)}$ is the corresponding record, formed by averaging the original pixel records over the disjoint blocks of $2^m \times 2^m$ pixels. In the usual way, each $L^{(m)}$ energy function comprises a likelihood contribution, or data fidelity term, and a prior contribution, or smoothness component. On the full pixel grid $L^{(0)}$, Jubb and Jennison use the energy function

$$H(\mathbf{X}) = \sum_{<s,t>} \beta \, I_{[\mathbf{X}_s \neq \mathbf{X}_t]} + (2\sigma^2)^{-1} \sum_{s \in L^{(0)}} (\mathbf{Y}_s - \mathbf{X}_s)^2,$$

where the notation $<s,t>$ denotes the two-pixel clique pair s and t, and $I_{[\mathbf{X}_s \neq \mathbf{X}_t]}$ is the indicator function taking the value 1 when $\mathbf{X}_s \neq \mathbf{X}_t$, and 0 otherwise. They choose to use the same form of the Markov random field model at all grid levels; the parameter β which represents the attractive strength between neighbouring pixels is held constant for all size pixels. If on the m^{th} level grid, the notation $<s,t>$ is used to denote pairs of adjacent pixels in $L^{(m)}$, then the energy function used by Jubb and Jennison can be written

$$H(\mathbf{X}^{(m)}) = \sum_{<s,t>} \beta \, I_{[\mathbf{X}_s^{(m)} \neq \mathbf{X}_t^{(m)}]} + (2\frac{\sigma^2}{2^{2m}})^{-1} \sum_{s \in L^{(m)}} (\mathbf{Y}_s^{(m)} - \mathbf{X}_s^{(m)})^2.$$

It might be considered that retaining the same functional form of the prior at the higher levels of the cascade introduces an undesirable inconsistency between levels in terms of the Markov random field construction. Consider Figure 5.1; this depicts the prior contributions involved in the first two levels of a cascade, the original pixel level, and the level after a 2×2 blocking. The prior is taken to have pairwise pixel interactions, including diagonal neighbours, and with the general form $\varphi_c(\,)$ replacing the specific $\beta I_{[\mathbf{X}_s \neq \mathbf{X}_t]}$ used above. The down-weighting given in Equation (3.9) will be used, namely a down-weight of $\sqrt{2}$ for diagonal neighbours. On the original level, shown in (a), the dark grey pixel contributes to the prior through its interaction with the eight light grey pixels, its Markov random field neighbours in $L^{(0)}$. Under the blocking scheme, the new Markov random field induces the neighbourhood scheme shown in (b). The dark grey and the three mid grey pixels form one new big pixel. This pixel generates prior contributions

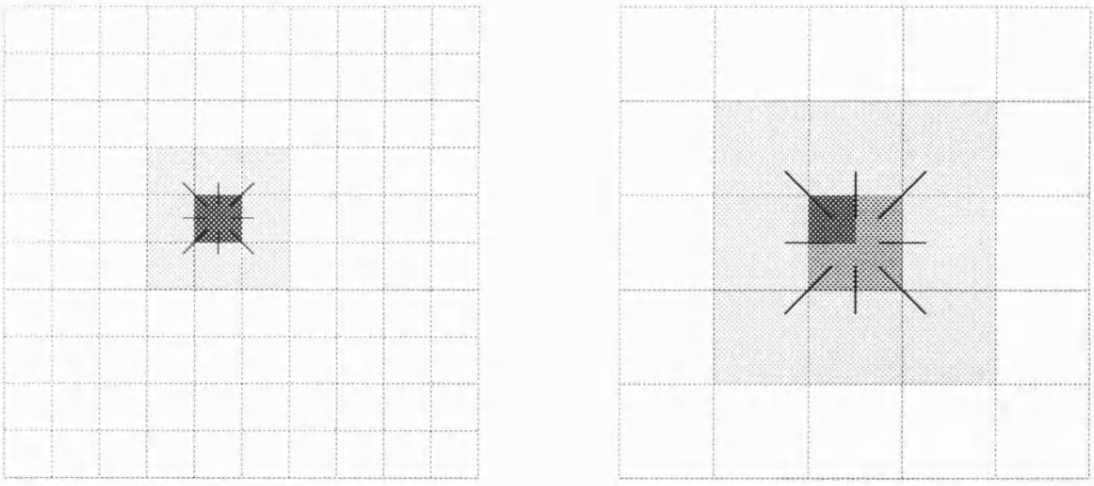


Figure 5.1 Original system of prior contributions.

on the same scale as before, through the interaction with its eight adjacent $L^{(1)}$ Markov random field neighbours. The neighbourhoods and the Markov random fields are not consistent between the cascade levels.

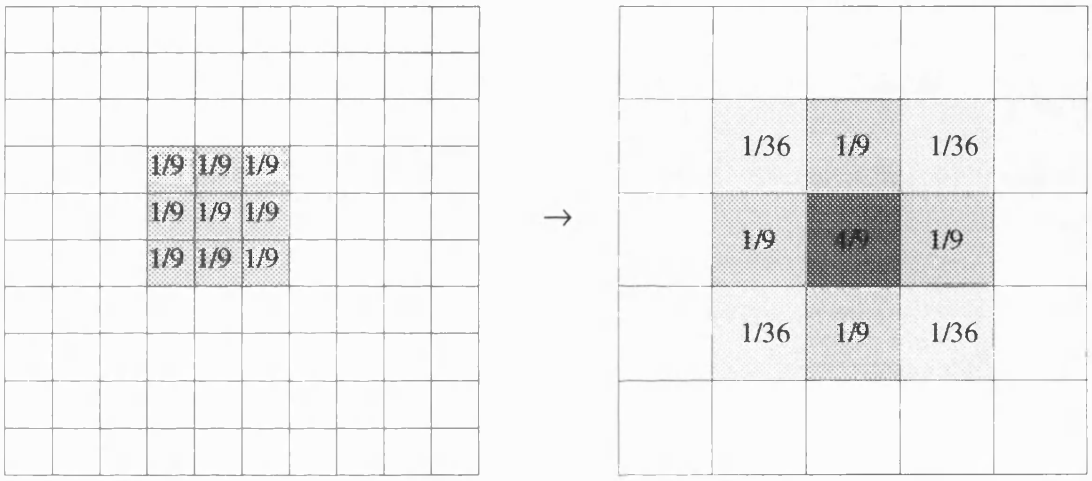
5.1.2 Inclusion of blurring

Jubb and Jennison only consider the application of cascade in restoring unblurred images where the data fidelity contribution to the energy is of the form $(2\sigma^2)^{-1}||\mathbf{Y}-\mathbf{X}||^2$. However in the general case, where blurring may occur, the fidelity contribution may be of the form $\lambda||\mathbf{Y}-\mathbf{KX}||^2$, where $\lambda=\lambda(\sigma)$ (see Section 2.1 for a discussion of the model formation, and Section 3.4.2 for the details of selecting λ for the models we have used).

Suppose that an attempt is made to include blurred images in the cascade framework by continuing to form the $\mathbf{Y}^{(m)}$ by averaging records over the $2^m\times 2^m$ blocks. The record formation on the original grid was modelled as $\mathbf{Y}_s=\sum_{t\in B_s}\gamma_{s-t}\mathbf{X}_t+\eta_s$, so

$$\begin{aligned}
\mathbf{Y}_j^{(m)} &= (2^{2m})^{-1}\sum_{s\in L^{(0)}}\mathbf{Y}_s^{(0)}, \qquad \text{where } s\in j^{th}\ 2^m\times 2^m \text{ block of } L^{(0)} \\
&= (2^{2m})^{-1}\sum_{s\in L^{(0)}}\Big(\sum_{t\in B_s}\gamma_{s-t}\mathbf{X}_t^{(0)}+\eta_s\Big) \\
&= (2^{2m})^{-1}\left\{\sum_{s\in L^{(0)}}\sum_{t\in B_s}\gamma_{s-t}\mathbf{X}_t^{(0)}+\sum_{s\in L^{(0)}}\eta_s\right\}.
\end{aligned}$$

In order to formulate a $||\mathbf{Y}^{(m)}-\mathbf{K}^{(m)}\mathbf{X}^{(m)}||^2$ for level m, the $\mathbf{X}^{(0)}$ components involved in the double summation, could be grouped into separate summations over the disjoint $2^m \times 2^m$ blocks of $L^{(0)}$ which constitute $L^{(m)}$. Unfortunately, on $L^{(m)}$, we only have the values of $\mathbf{X}^{(m)}$ available to use in the data fidelity term. Unless the components of $\mathbf{X}^{(0)}$ constituting any block take a common value, we will lose information in replacing each block of $\mathbf{X}^{(0)}$ by the corresponding $\mathbf{X}^{(m)}$ component. However if we do this, effectively constraining the block components to take a common value, some pseudo blurring coefficients $\gamma_{s-t}^{(m)}$ could be formed by averaging the corresponding γ_{s-t} block-wise. We will illustrate this idea with an example. Suppose we have an image degraded by 3×3 uniform blurring; the blurred value at any pixel s is the average of $\mathbf{X}_s$ with the value of its eight nearest neighbours. This is illustrated in Figure 5.2(a), where s is the central grey pixel.



(a) Original level $L^{(0)}$ (b) First cascade level $L^{(1)}$

Figure 5.2 Changes in example blurring coefficients due to 2×2 block averaging.

Consider averaging the records over a 2×2 block of $L^{(0)}$ pixels including s. Each of the four $L^{(0)}$ records will involve the values of nine of the $L^{(0)}$ pixels. In averaging the four records, we have a weighted average of the values of the 4×4 $L^{(0)}$ pixels centered on the 2×2 block. We could formulate an approximate weight for the contribution made by each pixel in $L^{(1)}$ to the averaged record, by averaging the γ_{s-t} corresponding to $L^{(0)}$ pixels lying in each 2×2 block. The resulting $\gamma_{s-t}^{(1)}$ in this case are as shown in Figure 5.2(b).

It seems that the effect of averaging records is to change the blurring structure at the different levels of the cascade. New pseudo blurring coefficients could be calculated, and the algorithm applied as before. However, these coefficients do not fully represent the way in which the records are formed.

5.2 Modifications to the cascade algorithm

5.2.1 Modifications to prior contributions

In Section 5.1.1, we described the inconsistencies between the Markov random fields defined at different cascade levels. In order to avoid these conceptual problems, we propose to regard the higher level $\mathbf{X}^{(m)}$, $m=1, \ldots, M$, as blocks of $L^{(0)}$ pixels constrained to take a common value, rather than as $L^{(m)}$ pixels per se. The prior contributions to the energy at any level m are then defined to be the prior contributions, under the original Markov random field model, from the $\mathbf{X}^{(0)}$ components consistent with the block values defined by $\mathbf{X}^{(m)}$.

The fact that pixels within blocks are constrained to take a single common value, by definition forming a first order region, suggests that a first order model may be appropriate (see Section 3.3). If this is the case, then using the first order cliques given in that section, namely nearest neighbour cliques, there is a simplification which can be made to the prior contributions. All contributions from cliques interior to any block will be zero since $D_c(\mathbf{X})=\mathbf{X}_s-\mathbf{X}_t$, and $\varphi_c(0)=0$. The contribution from a clique spanning separate blocks will be identical for all cliques spanning that particular block boundary. So, to calculate the entire prior contribution, we need only consider a single evaluation of all the between-block pixel contributions. This argument is illustrated in Figure 5.3 for a 2×2 aggregation when the model includes diagonal neighbours, and the down-weighting given in Equation (3.9) is used.

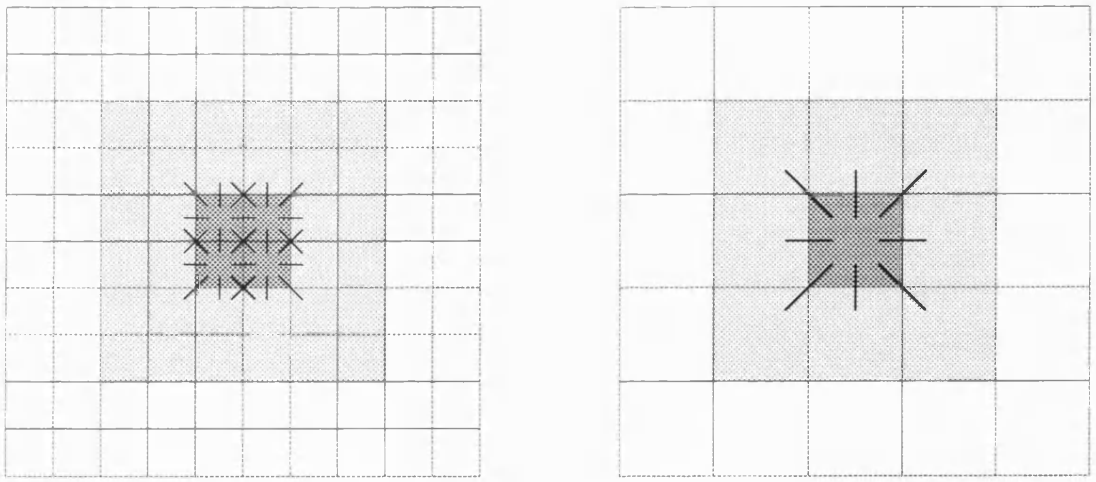


Figure 5.3 Redefined system of prior contributions.

Compare Figure 5.3(b) with Figure 5.1(b) which shows the corresponding values under the previous cascade definition. When the blocking is $2^m \times 2^m$, the scaling factors for the contributions are $2^m + 2(2^m - 1)/\sqrt{2}$ for horizontal adjacencies, and $1/\sqrt{2}$ for diagonal adjacencies (this assumes that diagonal adjacencies are to be used, if this is not the case, then the horizontal factor is simply 2^m).

It would be possible, when working with second or third order models, to constrain the values in each block of pixels to follow a single linear or quadratic equation. Again there would be a simplification in the total prior contribution, since within-block contributions would be zero. The between-block contributions would be different from the first order case as the cliques take a different form. Searching for the new block value would then require a consideration of all the possible planar or quadratic surfaces which could be fitted to the block, maintaining the range $\{0, \ldots, N-1\}$ at each pixel. In practise, this might be computationally infeasible, and for the remainder of this work we shall concentrate on first order models, with blocks of pixels constrained to take a single value.

5.2.2 Modifications to likelihood contributions

In Section 5.1.2, the problem of blurred records was discussed, and we saw that it was not possible to formulate the likelihood for the averaged records working with only the $\mathbf{X}^{(m)}$, $m=1, \ldots, M$. Given also that in the last section, we proposed regarding the higher level grids as blocks of constrained pixels, it seems reasonable to consider retaining all pixel records at all levels of the cascade. In this case, to update a block of pixels, the likelihood contributions would be considered from all the $L^{(0)}$ pixels which would have been involved had the $2^m \times 2^m$ pixels been simultaneously updated to their common value. So if on the original pixel grid, we have an energy function as defined in Equation (2.8),

$$\mathrm{H}(\mathbf{X}) = \sum_{c \in C} \varphi_c(\mathbf{X}) + \lambda \sum_{s \in S} (\mathbf{Y}_s - (\mathbf{KX})_s)^2,$$

then working on the $L^{(m)}$ grid, provided that the blocks of pixel values of $\mathbf{X}^{(0)}$ match the corresponding pixel values of $\mathbf{X}^{(m)}$, the $L^{(m)}$ energy can be written

$$\mathrm{H}(\mathbf{X}^{(m)}) = \sum_{c \in C} \varphi_c(\mathbf{X}^{(0)}) + \lambda \sum_{s \in S} (\mathbf{Y}_s^{(0)} - (\mathbf{KX})_s^{(0)})^2.$$

For the reasons discussed in Section 5.2.1, it is not necessary to calculate all the prior contributions in this case. The difference in energy between two realisations on $L^{(m)}$, $\mathbf{x}^{(m)}$ and $\tilde{\mathbf{x}}^{(m)}$, which differ at the single component T, can be written,

$$\sum_{\substack{c=\langle s,t\rangle \in C: \\ s \in \text{block } T \\ t \notin \text{block } T}} (\varphi_c(\mathbf{x}^{(0)}) - \varphi_c(\tilde{\mathbf{x}}^{(0)})) + \lambda \sum_{\substack{t \in S: \\ s \in B_t \ \& \\ s \in \text{block } T}} ((\mathbf{y}_t^{(0)} - (\mathbf{Kx})_t^{(0)})^2 - (\mathbf{y}_t^{(0)} - (\mathbf{K}\tilde{\mathbf{x}})_t^{(0)})^2).$$

The non-averaging approach to the records can be shown to be equivalent to the Jubb and Jennison approach in the unblurred case they considered. Consider the contributions to the likelihood from the T^{th} pixel of $L^{(m)}$, and let s index the $L^{(0)}$ pixels which correspond to this block. Suppose the $L^{(0)}$ pixels in the block T are constrained to take a common value, the value of $\mathbf{X}_T^{(m)}$, then

$$\begin{aligned}
\sum_{s \in \text{block } T} (\mathbf{y}_s^{(0)} - \mathbf{x}_s^{(0)})^2 &= \sum_{s \in \text{block } T} (\mathbf{y}_s^{(0)} - \mathbf{x}_T^{(m)})^2 \\
&= \sum_{s \in \text{block } T} (\mathbf{y}_s^{(0)} - \mathbf{y}_T^{(m)} + \mathbf{y}_T^{(m)} - \mathbf{x}_T^{(m)})^2 \\
&= \sum_{s \in \text{block } T} ((\mathbf{y}_s^{(0)} - \mathbf{y}_T^{(m)})^2 + 2(\mathbf{y}_s^{(0)} - \mathbf{y}_T^{(m)})(\mathbf{y}_T^{(m)} - \mathbf{x}_T^{(m)}) + \\
&\qquad (\mathbf{y}_T^{(m)} - \mathbf{x}_T^{(m)})^2) \\
&= \sum_{s \in \text{block } T} (\mathbf{y}_s^{(0)} - \mathbf{y}_T^{(m)})^2 + 0 + 2^{2m} (\mathbf{y}_T^{(m)} - \mathbf{x}_T^{(m)})^2,
\end{aligned}$$

since the sum over the block of pixel records $\mathbf{y}_s^{(0)}$ equals the sum over the block of their average $\mathbf{y}_T^{(m)}$. The contribution to the energy from the $L^{(m)}$ pixel T would be $(2\sigma^2/2^{2m})^{-1}(\mathbf{y}_T^{(m)} - \mathbf{x}_T^{(m)})^2$. The contribution to the energy from the 2^{2m} $L^{(0)}$ constrained pixels in block T would be $(2\sigma^2)^{-1} \sum_{s \in \text{block } T} (\mathbf{y}_s^{(0)} - \mathbf{x}_T^{(m)})^2$. We have just shown that these two expressions differ only by a term which is independent of the current value of any of the $\mathbf{X}$, and so will not be used in the restoration algorithms. The equivalence is due to the fact that the observations are assumed to be independent and identically Normally distributed, so that the average record is then a sufficient statistic for the mean of the distribution.

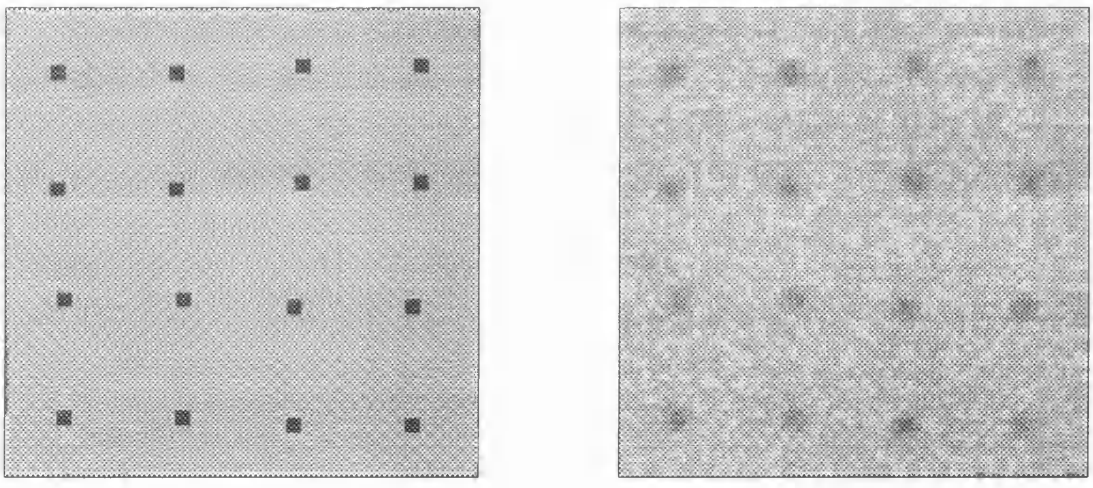
5.2.3 Connection between different levels

The cascade algorithm, following the modifications we have suggested, could be viewed in two ways. In the case without blurring, as is considered by Jubb and Jennison, we have seen that our approach is equivalent to their approach using a reweighted system of Markov random field interactions (see Figure 5.3). The higher grids could then be considered as large pixels, each with a record formed by averaging the true records. Alternatively, with or without blurring, the algorithm could be regarded as initialising processing on an image constrained so that large blocks of pixels take a common value. As the level changes, constraints are removed to allow the blocks to subdivide into smaller blocks which can be updated separately, eventually reaching the original level, where pixels are updated individually. The energy of an $L^{(m)}$ scene is identical whether calculated from the values of the original pixels, or from the values of the block pixels.

As a result of these proposed changes, the connections between the energy functions at different cascade levels are clear. At higher levels of the cascade, we are working with a subset of all the possible images in Ω. We are restricted to the section of the full $L^{(0)}$ energy function corresponding to this subset of images. Updating a block of pixels corresponds to taking a large step across the energy function between two of these permitted images. At each lower level of the cascade, more images can be realised, and this increases the number of points in the energy function which can be visited. At the lowest level, where individual pixels are updated separately, we can reach all points in the original energy function. Since the minimisation algorithm, ICM or simulated annealing, is applied to the energy function at each level, it seems reasonable to provide a sequence of energies layered in this way.

5.2.4 Examples on test scene

In order to assess the effectiveness of the algorithm, we will present results for three test scenes. These scenes are designed to test the method, and to highlight any weaknesses which may suggest further modifications. All three images have 64×64 pixels taking 64 possible grey-levels, and each consists of sixteen 2×2 foreground blocks in a uniform background. Only one of the original images is shown, Figure 5.4, since the differences are not visually significant. It should be noted that for display, the grey-level scale has been stretched to emphasize features (as described in Section 3.4.4); all these scenes, their records, and any reconstructions are displayed on this common scale.



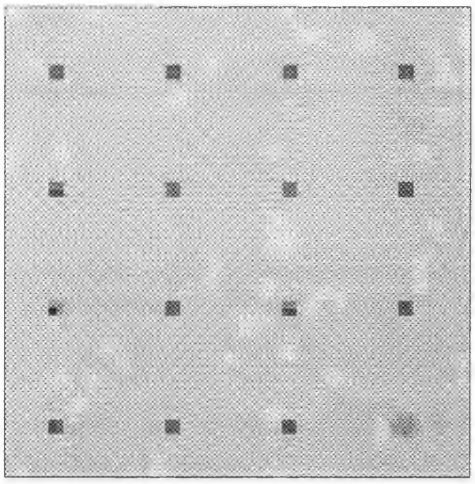
(a) (b)

Figure 5.4 (a) Test image containing eight aligned and eight non-aligned blocks, and (b) record after 3×3 uniform blur and $N(0,1)$ noise.

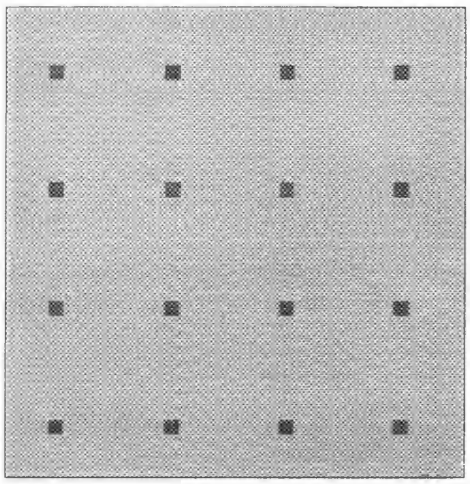
In the first test scene, the 2×2 pixel foreground objects are positioned on the pixel grid so that they are aligned with the 2×2 blocks formed at the $L^{(1)}$ level of the cascade. In the second test scene, all the foreground objects are displaced by one pixel in both the horizontal, and the vertical directions. The third test scene is a patchwork of alternate quarters of the previous two, so that eight foreground objects are aligned, and eight are non-aligned. It is this final image which is shown in Figure 5.4(a). The record for the third image after degradation by uniform 3×3 blurring and $N(0,1)$ noise is shown in Figure 5.4(b). Under a four neighbour first order Geman and Reynolds' model with parameters $\Delta=10$ and $d=3$, as described in Chapter 3, the energy of the test scene is 429.9. These three test scenes all have the same energy value under this model since all three of the images contain the same number of discordant pixel clique pairs, and their records are all generated with the same simulated Normal noise.

Figure 5.5 shows six reconstructions, two for each of the test images. The first reconstruction of each image employs standard single-site update ICM. The second reconstruction uses the modified cascade described in Section 5.2 to apply ICM to the different block sizes of pixel. For the moment, we are restricting the application of cascade to use with ICM. Although they are not shown, reconstructions were also obtained with Gibbs sampler based simulated annealing, using a linear schedule decreasing from temperature 1 to 0 in 100 sweeps. Simulated annealing is a stochastic algorithm, and strictly speaking it is insufficient to present comparisons based on a single outcome from the chain. Reconstructions will vary depending on the samples drawn for the proposal distribution at each update. These samples are generated using a pseudo random number generator, and so the generator seeds will determine the outcomes. The simulations in Stander (1992) appear to demonstrate that the amount of variability in the final energy value will be small for a choice of schedule which gives a low final energy. In practise, it may be sensible to use the computational resources available to identify a good schedule to apply for a number of reconstructions, rather than to generate a repeated series of reconstructions for each problem under some less optimal schedule. We have restricted ourselves to presenting a single reconstruction in each case; it should be remembered that the final reconstruction energies quoted for simulated annealing do have an associated variability.

Now consider the reconstructions, and their corresponding energies, shown in Figure 5.5. For comparison, the simulated annealing reconstructions have energies 437.8, 436.0 and 436.6 respectively; visually, they are fairly successful at recovering the foreground objects. Standard ICM appears to suffer difficulties in identifying both the foreground, and the background levels. The cascade reconstructions do all have lower energies than their single-site update counterparts;

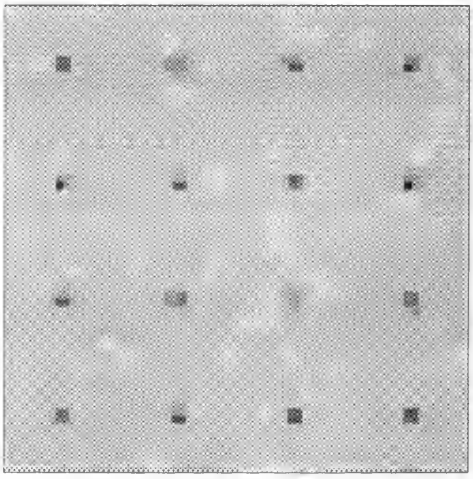


(a) ICM: Energy = 476.5

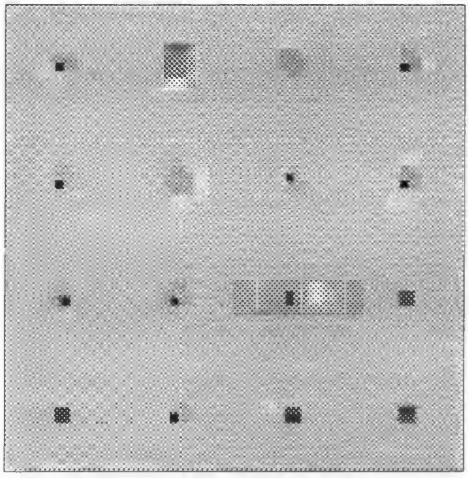


(b) Cascade ICM: Energy = 428.2

First test image - aligned blocks

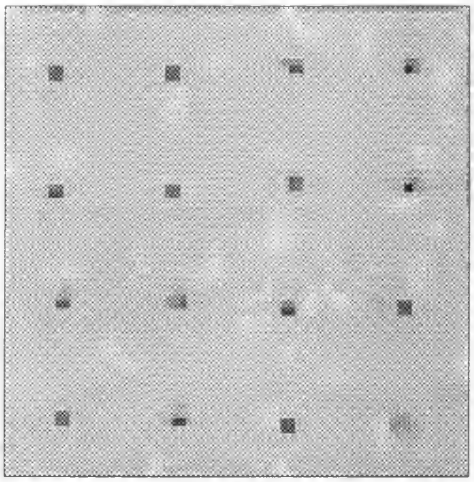


(c) ICM: Energy = 501.5

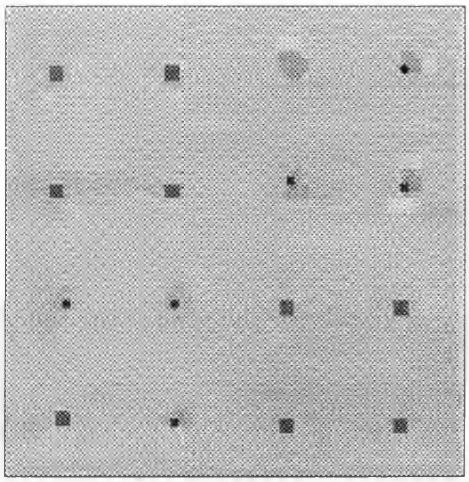


(d) Cascade ICM: Energy = 479.7

Second test image - nonaligned blocks



(e) ICM: Energy = 487.2



(f) Cascade ICM: Energy = 454.0

Third test image - aligned and nonaligned blocks

Figure 5.5 Reconstructions of test images, four neighbour model with Δ=10, d=3.

this appears to be due in part to the improved reconstruction of the background pixels. As might be expected, the cascade is far more successful when the blocking scheme and the objects coincide; visually, the non-aligned blocks are badly identified. It should be noted that while reconstructions using ICM take a comparable amount of CPU time with, or without, cascade, simulated annealing takes approximately thirty times longer in this implementation.

5.3 Adaptive cascade

5.3.1 Introduction and aims

The reconstructions in Figure 5.5 suggest that cascade can improve the performance of ICM in tackling certain problems of this simple type. Unfortunately, the reconstructions also demonstrate the susceptibility of the algorithm to slight perturbations in the specific test scene. In the discussion in Chapter 4, we stated that the motivation for cascade was to provide a sequence of minimisation problems to tackle, the solution to each of which should provide a good starting value for the following problem. The algorithm is not intended to provide better sampling behaviour at fixed temperatures. The two aims are quite distinct; whilst a good sampler should be sufficiently mobile to visit all areas of the density in the correct proportions, the search path for the minimisation problem may need to be more directed.

In our revised definition of the cascade algorithm, the large pixels of Jubb and Jennisons' formulation are to be considered instead as groups of pixels constrained to take a common value. Under this definition, we are no longer confined to partitioning the scene into square blocks. The images attainable at the higher levels of the cascade correspond to nested subsets of the entire set of possible images. These subsets are obviously determined by the choice of blocking scheme used in the cascade. As the examples in the previous section have demonstrated, the naive quartering scheme may not be optimal in terms of the final energy attained. We might expect that a scheme which can identify groups of pixels with similar behaviour might be able to generate a more appropriate sequence of these subsets.

There are many possible approaches to partitioning the image, and these will obviously vary in their computational complexity. For images such as the current test scenes, consisting of distinct, identical objects on a uniform background, it would probably be possible to devise some good, ad hoc blocking scheme. However, there are two conditions which it seems reasonable to impose on any potential blocking scheme. The first condition is that the scheme is robust in some way, that is, it is not so specialised that it performs very well on a small class of images, but very badly outside this class. The second condition is that the

scheme is not too computationally demanding, either in terms of the CPU time required to implement the scheme, or the complexity required in programming. Cascade is intended to be a conceptually simple approach which capitalises on the computing resources available. Additionally, as we will discuss in Chapter 6, a stochastic algorithm such as simulated annealing may already be more robust to the blocking than is a deterministic algorithm such as ICM.

In line with Jubb and Jennisons' cascade, we will follow the same pattern of increasing the number of regions at consecutive levels by a factor of four; we begin with a four region scene, move to a sixteen region scene, and so on. The initial test partitions are of a simple nature; the basic extension is to permit rectangular, rather than strictly square, blocks of pixels. To do this, we will partition the pixel grid row-wise and column-wise. The choice of the partitions will be data-driven.

5.3.2 Partitioning scheme

In order to partition the image, we will require some measure of the dissimilarity between pixels potentially lying in different blocks. In the context of boundary detection for textured scenes observed without noise or blur, Geman, Geman, Graffigne & Dong (1990) suggest two statistics to test the disparity between two blocks of pixels. In their problem, the blocks are already determined, the decision is whether adjacent blocks should be separated by an edge (see Section 3.2.2 for a brief discussion of edge processes). The first statistic they suggest is a Kolmogorov-Smirnov measure involving the sample distribution functions of the (possibly transformed) pixel values in the two blocks. Under the null hypothesis that the pixels in the two blocks are from the same continuous underlying distribution, the probability density of this statistic is independent of the underlying distribution. The second statistic, which they use as an alternative when the blocks consist of a single pixel each, is a weighted absolute difference between the pixel values (the weighting is chosen to make the statistic invariant to linear transformations of the pixel values). With either statistic, a hypothesis test can be carried out to determine whether, or not, a boundary should be present between the two blocks.

Our partitioning scheme does bear some resemblance to the above situation. We are interested in identifying groups of pixels which may be exhibiting different behaviour, or equivalently blocking together those which are exhibiting similar behaviour. However, our problem is firstly one of selection. At each new level of the cascade, we wish to increase the number of regions from n to $4n$. In our simple initial efforts, this amounts to selecting the appropriate number, $\sqrt{n}$, of paired row-wise, and column-wise splits of the pixel grid. So we are interested in a ranking of the disparity statistics from the potential new partitions, the associated

significance levels are of less relevance. Since we are also looking for a simple scheme, we will consider adapting, in some way, a statistic based on the absolute difference in pixel records between blocks.

Before we discuss the form of the statistic we propose to use, we should mention the simplifications which will be made, mainly for computational reasons. Suppose we wish to increase the number of blocks from n to $4n$. If we assume the pixel grid is square, dimension $|S^0|^{1/2} \times |S^0|^{1/2}$, then the division into n blocks is specified by a paired list of $\sqrt{n}-1$ row-wise and column-wise, partitions from the $|S^0|^{1/2}-1$ possibilities. In order to simultaneously select the additional $\sqrt{n}$ row-wise and column-wise, partitions which specify the division into $4n$ blocks, we need to tackle two problems. The first is computational; consider the number of possible ways in which we could simultaneously choose the partitions. This equals the number of ways of selecting $\sqrt{n}$ partitions from the $|S^0|^{1/2}-\sqrt{n}$ remaining possibilities, then squared to take into account both rows and columns. At higher levels of the cascade, this number could be prohibitively large. The second difficulty is the question of combining the disparities between all the blocks created by a potential new partitioning. Geman, Geman, Graffigne and Dong only consider the pixel differences across a single block boundary. We would need to consider the total effect of the new blocking scheme, that is a combination of the disparities between all adjacent blocks. Using a statistic involving the difference in records across boundary blocks, these disparities would not be independent. For these reasons, we will select row and column partitions independently, with the disparity only calculated across the new partition. So, in selecting a horizontal partition, the position of existing horizontal partitions will be taken into consideration, but vertical partitions will be ignored (and vice versa). In effect, we are considering the differences between strips of the image. In addition, in selecting either set of partitions, we will adapt a sequential, rather than a simultaneous, procedure. The disparity statistics will be calculated for the $|S^0|^{1/2}-\sqrt{n}$ potential sites. The site giving the most extreme value is selected, and a partition is then considered to exist at this site. The statistics for the remaining sites can then be recalculated, if necessary, and again the most extreme site selected, and so on. It is hoped that the cycles of calculation, followed by selection of the most extreme value, may be nearer in final outcome to a simultaneous evaluation than would be a selection of the $\sqrt{n}$ most extreme values from a single calculation.

We can now consider the form of the dispartity measure to use. Denote a statistic for site i, between rows or columns i and $i+1$, by Q_i. The statistic Q_i is to be based on the difference in average record of the two pixel strips created between the proposed partition in position i, and it's enclosing partitions on either side. For this definition, the edges of the image are considered to be partitions.

The number of pixels in these strips will vary depending on the distance from existing partitions. However, the assumption has been made that the record noise is Normally distributed, with common variance σ^2. So, for comparison, the $\{Q_i\}$ could be normalised to have common variance σ^2. Denote the pixel strip to one side of the partition by L_i, containing $|L_i|$ pixels for which records exist, and the pixel strip on the other side by R_i. The variance of the average record in L_i will be $\sigma^2/|L_i|$, and the variance of the difference between the two average records will be $\sigma^2(1/|L_i|+1/|R_i|)$. Provided that both strips contain at least one record, we could define a disparity statistic to be

$$Q_i = \Big(\sum_{s \in L_i} \mathbf{Y}_s /|L_i| - \sum_{t \in R_i} \mathbf{Y}_t /|R_i| \Big) / \big(1/|L_i| + 1/|R_i| \big)^{1/2}. \qquad (5.1)$$

When either $|L_i|$ or $|R_i|$ is zero, the convention would be to set Q_i equal to zero. Since we will select a site i if it has the largest absolute value of Q_i over the potential i, this convention will result in the edge sites being selected last. This seems reasonable since we have less information about the corresponding pixels.

The statistic Q_i given in Equation (5.1) should be amenable to calculation, and simplifications could possibly be obtained by storing row and column totals, together with the number of records in each total. However, at high levels of the cascade, the blocks may be very large in relation to the size of feature in the image. The result of averaging records over a large number of pixels may be to overwhelm any effect due to small features, for example the 2×2 foreground blocks in our test examples. For this reason, a slight modification of the defined Q_i will be considered.

Suppose that we retain the general form given in Equation (5.1), but redefine the two strips L_i and R_i. We will introduce a new, positive parameter ξ. Then, rather than averaging all the records between the potential partition and the existing enclosing partitions, we average just the first ξ rows on either side. If an existing partition is encountered before ξ is counted out, then the averaging is truncated at this point after ξ_i rows on both sides; otherwise, ξ_i equals ξ,

$$Q_i^{\xi} = \Big(\sum_{s \in L_i^{\xi}} \mathbf{Y}_s /|L_i^{\xi}| - \sum_{t \in R_i^{\xi}} \mathbf{Y}_t /|R_i^{\xi}| \Big) / \big(1/|L_i^{\xi}| + 1/|R_i^{\xi}| \big)^{1/2}. \qquad (5.2)$$

The intention is to choose the parameter ξ to be of the order of the features of interest in the scene. At low levels of the cascade where there are already many partitions, or when ξ is large, the results using this Q_i^{ξ} should be approximately the same as those using the original Q_i. However, at higher levels, we hope that the partitioning will be more sensitive. We will use the form given in Equation (5.2) for the examples we will show; both forms have been implemented, and the latter did appear to perform better with our test examples.

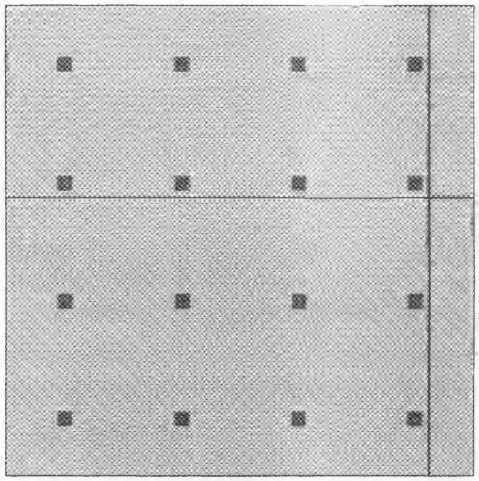
5.3.3 Examples

In this section, we will demonstrate the adaptive cascade which we have just described, applied to the second and third test examples used in Section 5.2.4. These two test scenes both contain sixteen foreground blocks; in the second test example these are badly positioned for a standard cascade, in the third we have a mixture of the two block positions. Figures 5.6 and 5.7 depict adaptive cascades applied to the second and third test scenes respectively. In order to assess the effectiveness of the partitioning scheme at different cascade levels, images (a)-(e) of both figures show the appropriate clean test scene overlaid by the block boundaries at a sequence of cascade levels. The image (f) of both figures is the final reconstruction following the adaptive cascade using Q_i^{ξ} given by Equation (5.2) and with $\xi=2$. For comparison with the earlier examples, the minimisation technique employed is ICM, and a four neighbour Geman and Reynolds model has been used with $\Delta=10$ and $d=3$.

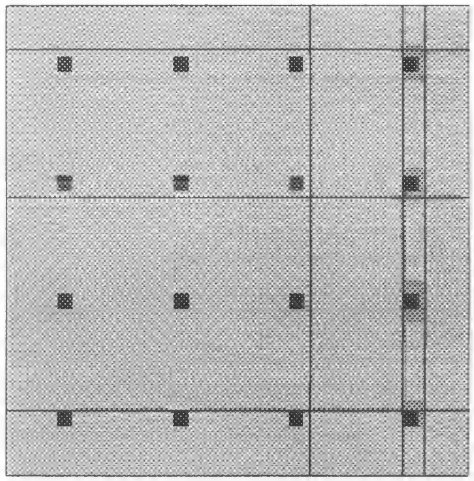
We will first consider Figure 5.6, which involves the test scene designed to be awkward for a standard cascade. By comparison with the usual cascade reconstruction, shown in Figure 5.5(d), there is an improvement in the final energy attained, from 479.7 to 440.2. Visually, the reconstruction is also more successful, recovering the majority of the foreground blocks. Applying the same form of the adaptive cascade to the first test scene, yields an energy of 429.0, only slightly worse than the original cascade reconstruction shown in Figure 5.5(b) which has an energy of 428.2; this reconstruction is not shown.

Although the adaptive cascade is giving an improved final reconstruction, regarding Figure 5.6(a)-(e), it is clear that the partitioning is not working quite as might be expected. In particular, the partitions frequently seem to occur one pixel away from the edge of the 2×2 objects, this is apparent in (a)-(d). In order to investigate this occurrence, we will consider the case of a single 2×2 block of pixels taking some foreground value J in an otherwise large uniform background taking the value zero. The choice of levels 0 and J is without loss of generality for steps of size J, since it is only the difference in pixel value which is used. If a 3×3 uniform blurring is applied, we obtain the following scene

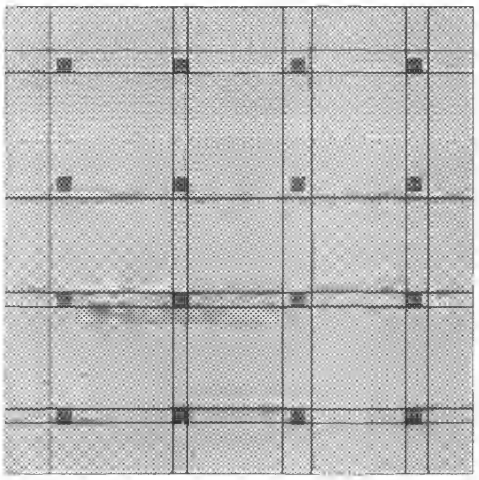
$$\mathbf{X}: \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & J & J & 0 & 0 \\ 0 & 0 & J & J & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \rightarrow \mathbf{KX}: \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & J/9 & 2J/9 & 2J/9 & J/9 & 0 \\ 0 & 2J/9 & 4J/9 & 4J/9 & 2J/9 & 0 \\ 0 & 2J/9 & 4J/9 & 4J/9 & 2J/9 & 0 \\ 0 & J/9 & 2J/9 & 2J/9 & J/9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & \uparrow A & \uparrow B \end{matrix}.$$



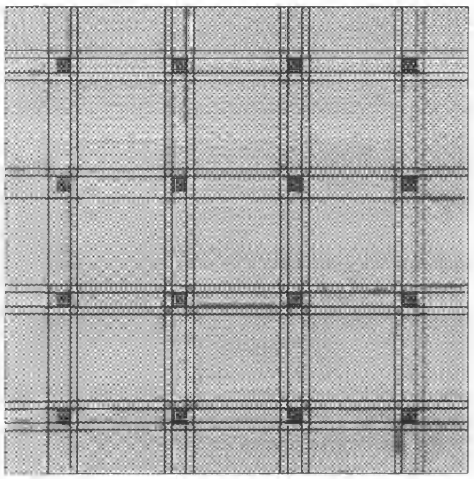
(a) Fifth level partition



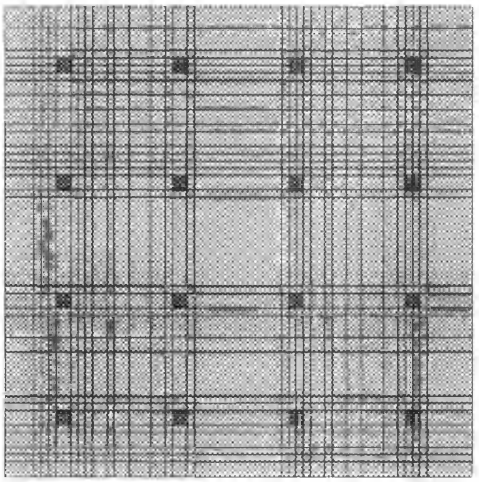
(b) Fourth level partition



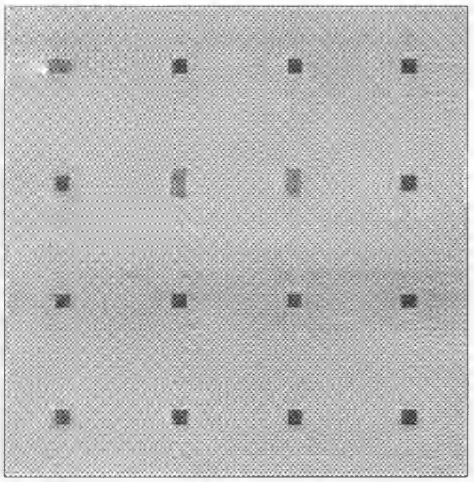
(c) Third level partition



(d) Second level partition

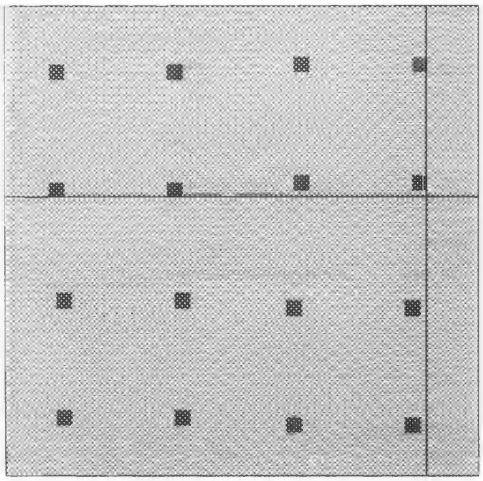


(e) First level partition

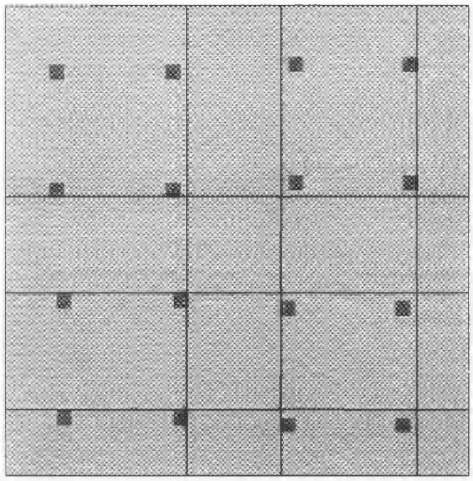


(f) Reconstruction: Energy = 440.2

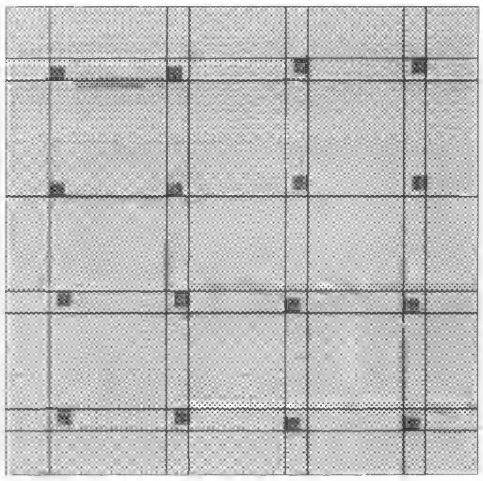
Figure 5.6 The second test scene overlaid by $\xi=2$ partitions at different levels, and the cascade ICM reconstruction using a four neighbour model, $\Delta=10$ and $d=3$.



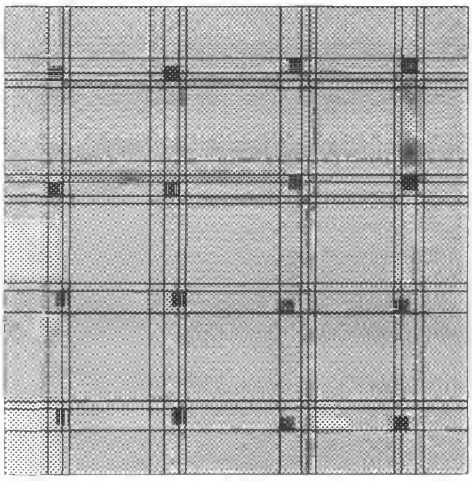
(a) Fifth level partition



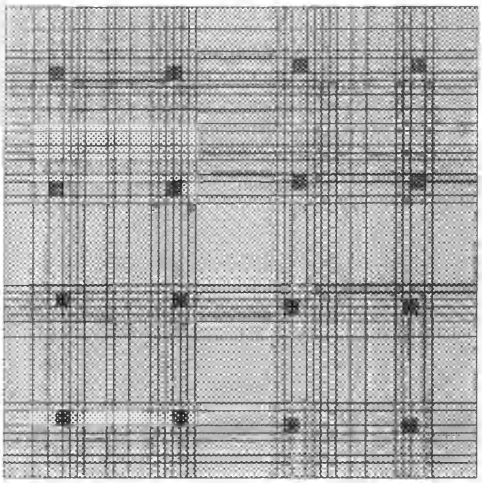
(b) Fourth level partition



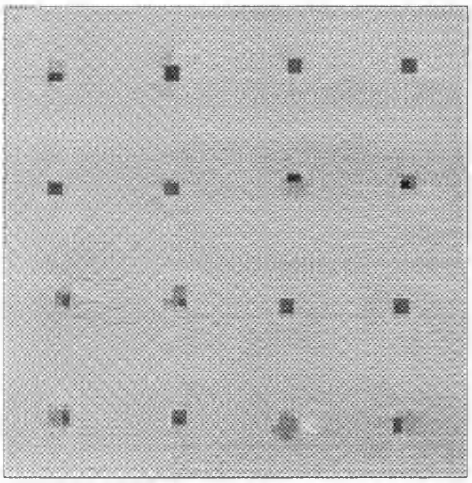
(c) Third level partition



(d) Second level partition



(e) First level partition



(f) Reconstruction: Energy = 451.7

Figure 5.7 The third test scene overlaid by $\xi=2$ partitions at different levels, and the cascade ICM reconstruction using a four neighbour model, $\Delta=10$ and $d=3$.

The corresponding records would then be these blurred values plus some Normal noise with mean zero. In this case, the expected column totals for the four central columns are $2J/3$, $4J/3$, $4J/3$ and $2J/3$, all other expected column totals are zero. Suppose a vertical partition is to be formed with $\xi=2$, and in the absence of any existing partitions. It can be seen that for positions *A* or *B*, the expected difference of two column totals either side of these partition sites is $2J$. So, the choice between these two partitions will be determined by the particular realisation of the noise. The problem is not resolved by altering ξ; with $\xi=1$, there is the same dependence on the noise realisation. With larger ξ, the problem is worse, with position *B* favoured.

It is worth noting that this effect might not occur if the foreground object was at least the size of the support of the blurring (in this case 3×3). It is also a feature of uniform blurring, as can be seen from a 1-dimensional example. Suppose we have some blur represented by the coefficients $\{\gamma_{-1}\ \gamma_0\ \gamma_1\}$, $(\gamma_{-1}+\gamma_0+\gamma_1=1)$, acting on the sequence of values $\cdots, 0, 0, 1, 1, 0, 0, \cdots$. The resulting non-zero section of the blurred sequence is γ_1, $\gamma_0+\gamma_1$, $\gamma_{-1}+\gamma_0$, γ_{-1}. If $\xi=2$, then the expected difference for a partition correctly positioned at the left of the block is $2\gamma_0+\gamma_{-1}$. The expected difference for the partition one position further from the block, at the extreme of the effect of the block in the blurred values, is $2\gamma_1+\gamma_0$. The former, correct position will therefore be favoured when $2\gamma_0+\gamma_{-1}>2\gamma_1+\gamma_0$. Using the fact that $\gamma_{-1}+\gamma_0+\gamma_1=1$, this condition reduces to $\gamma_1<1/3$. A similar argument for the right of the block gives the condition $\gamma_{-1}<1/3$.

It was considered whether some weighting could be introduced into Q_i^{ξ}, weighting paired differences of column totals equidistant from the partition *i*. The weights would be chosen to favour a partition correctly identifying the object boundary. However, it was decided that any such weights would be too problem specific, heavily dependent on the form of the blurring and the size of the object; a non-test scene might reasonably be expected not to consist of identical isolated foreground objects in a uniform background. Another problem to note is that although our test examples were fairly robust to the choice of ξ, the best results were obtained for $\xi=2$. Without prior knowledge of the scene, the use of a single ξ may be questionable.

The relative success of the adaptive partitioning scheme applied to the second test scene is not particularly surprising given the geometry of the objects. It is more interesting to see how the partitioning scheme copes when the object edges are not so conveniently positioned. This is the case for the third test scene, and the results of this partitioning are shown in Figure 5.7. In this case, the reconstructions are comparable with those produced by the standard cascade, shown in Figure 5.5(f), with only slight improvements both visually, and in the

energy from 454.0 to 451.7. It does not seem likely that a different choice of Q_i^{ξ} could dramatically improve this performance; it may even be that this particular Q_i^{ξ} is optimal for this situation, since the one-pixel-offset effect described above, results in the best choice under the circumstances. The problem in this case lies in the restrictive nature of partitioning the entire scene row-wise and column-wise.

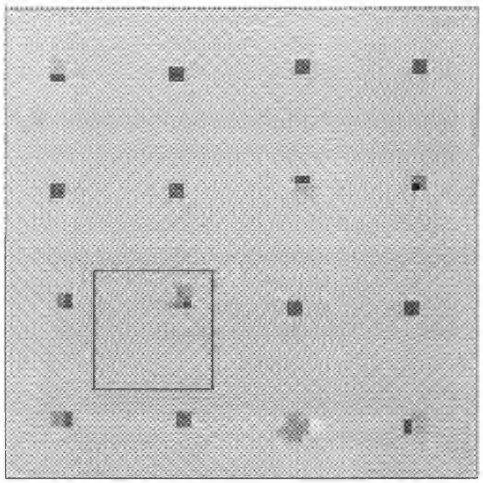
These two test examples seem to have identified two difficulties with the partitioning scheme as it stands: the identification of a suitable ξ, and the inadequacies of the crude blocking system. It seems likely that both of these problems could be solved, although possibly with not insubstantial computational effort. However, there is one relatively simple improvement which can be implemented, a further extension of an adaptive cascade, and this will be described in the next section.

5.4 Further implementation extensions

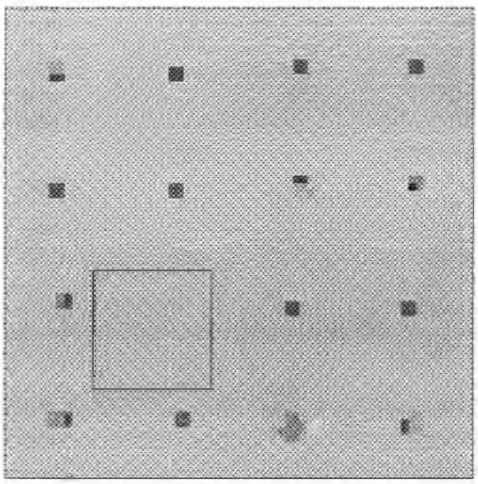
5.4.1 A window cascade

One of the problems of the adaptive cascade as described so far, is that the partitioning is on too global a scale. The results might be better if the partitions could be formed on a more flexible basis, capturing local features. A partitioning scheme capable of dividing the image into an appropriate number of polyhedral pixel blocks would obviously be ideal. However, we have already discounted this idea, at least for the present time, on the grounds of its complexity in terms of formulation and implementation. One simple way to achieve increased local input to the cascade is to restrict the partitioning, and updating, to within a small region, or window, of the scene. This general idea is illustrated in Figure 5.8; we will give the details of the method below, before discussing this example.

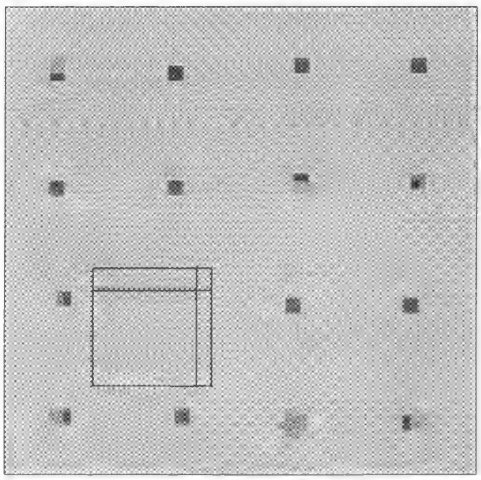
The intention is to obtain an initial reconstruction by some method, cascade or non-cascade. A small window of the image is then chosen in some systematic way from the pixel grid. At each position of the window, the whole process of an adaptive cascade is run strictly within the window. The starting value for the cascade is taken to be the grey-level closest to the average of the current pixel values within the window. The standard partitioning described in Section 5.3.2 is then applied, with the pixel strips containing only records within the window, in order to produce a partition of the within-window scene. At the updating stages, all pixel values outside the window are held fixed. Each within-window block is updated in the usual way, using all the relevant pixel values and records. In this implementation, the revised within-window reconstruction, after the cascade has been completed, is only accepted if this would lead to a reduction in the energy of the entire scene. However, it would be possible to introduce some properly formulated stochastic acceptance rule to permit certain increases in energy.



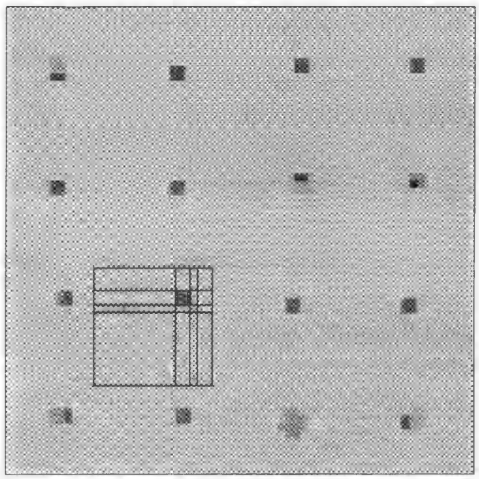
(a) Adaptive cascade reconstruction
Energy = 451.7



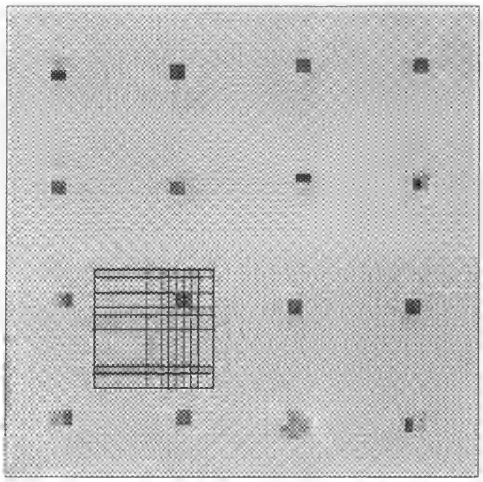
(b) Average pixel value
within window



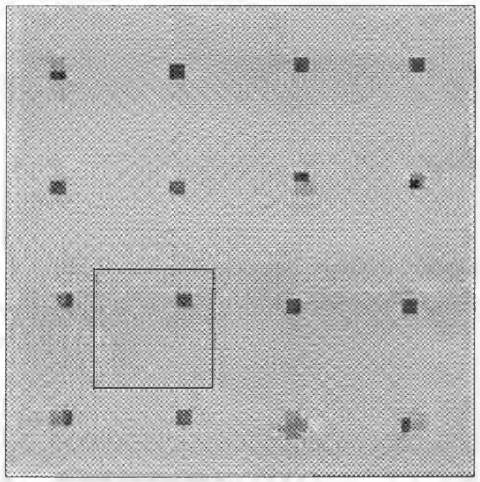
(c) Reconstruction at third
level, plus partitions



(d) Reconstruction at second
level, plus partitions



(e) Reconstruction at first
level, plus partitions



(f) Final reconstruction
Energy = 448.4

Figure 5.8 The stages of one window cascade ξ=2 for the third test scene.

Suppose we wish to run an adaptive cascade with some particular ξ within the windows. The choice of ξ is related to the scale of feature in the scene in which we may be interested. Since the intention of the window cascade is to improve the restoration of such features, it seems reasonable to link the size of the window to ξ. The window should be small enough that the partitioning is local in terms of the order of ξ, and features are not swamped. However, the windows should also be large enough that the averaging of the records reduces the noise variance sufficiently that the effect of order ξ features is noticeable in the statistic Q_i^{ξ}. As a compromise, we have chosen the windows to be square with sides of length four times the smallest integer power of 2 greater than ξ; the use of powers of 2 simplifies the computing. If the dimensions of the window resulting from this rule become as large as those of the entire scene, the window dimensions are halved. For example on a 64×64 pixel grid, when ξ=2 or 3, the windows are 16×16, but if $\xi \geq 4$, then the windows have dimension 32×32.

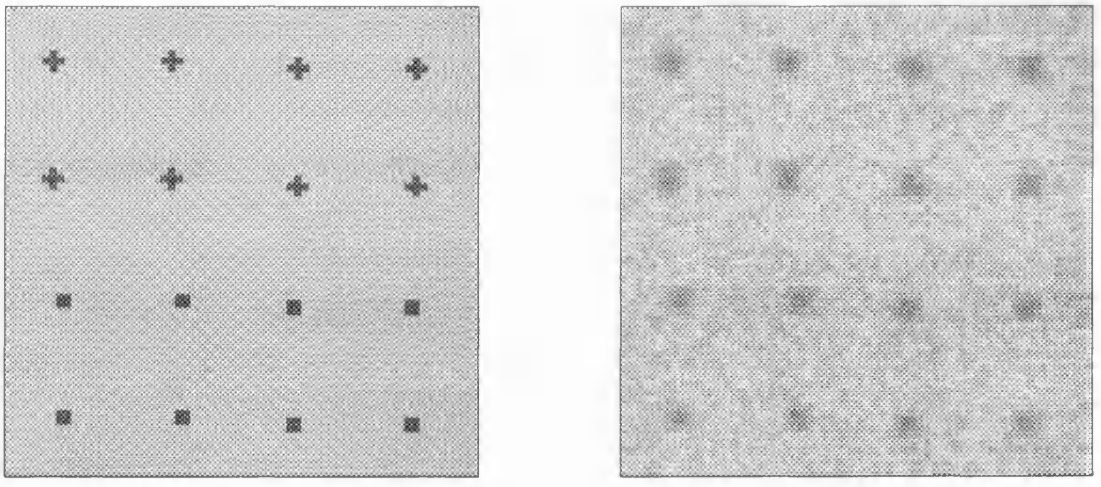
In terms of moving the window across the image, it seems reasonable to expect that better results would be obtained if an idealised size $\xi \times \xi$ feature, such as in our test examples, were to lie entirely within at least one window. For this reason, in moving the window to a new position, we have chosen to overlap the two positions by one quarter of the number of pixels in the window; from our earlier choice of the window size, this overlap is then greater than ξ, provided ξ is not too large. The scheme we have used is to shift the window across the scene horizontally, with the overlap determining the new position. Once the entire width has been covered, we move back to the beginning of the row, but at a new vertical position, again determined by the overlap. This system may not require an integer number of windows to cover the image. We have chosen to use one extra complete window, absorbing the excess covering width by splitting it as a larger overlap at the edges of the pixel grid.

We can now consider Figure 5.8 again. This shows the stages of an adaptive cascade, run within a single window, for the third test scene which so far has proved difficult to reconstruct with cascade. The initial starting reconstruction is the ξ=2 adaptive cascade carried out over the entire pixel grid. Comparing (a) and (f), it is clear that the window cascade is fulfilling its purpose for this test example. The final reconstruction after a series of window cascades, following the scheme we have described, has not been shown, however it has the lowest energy achieved as yet for this example, 433.3 (compare this with 487.2 for standard ICM, 436.6 for standard simulated annealing, 454.0 for standard cascade, and 451.7 for an entire pixel grid adaptive cascade). Similarly, following the same procedure, the energies for the first and second test scenes are 428.2 and 430.2; these are the lowest, or joint lowest with standard cascade for the first test scene, so far.

An advantage of the window idea is that we may be able to compensate for the limitations of a particular partitioning scheme by sequentially applying a number of slightly modified schemes, each within its own window cascade. For example, we are not restricted to a single value of ξ; we could run a sequence of window cascades across the scene, each using a different ξ. Also, although we described the adaptive cascade for a partitioning scheme which generates rectangular blocks, and then formulated the window cascade to run this scheme within square windows, these are just particular examples of what could be done. Our redefinition of cascade allows far more general forms for both the partitioning and the window shape. One possible extension, which will be discussed in the next section, is a cascade which partitions diagonally across the pixel grid.

5.4.2 A diagonal cascade

The adaptive cascade schemes considered so far have concentrated on horizontal and vertical partitions. However, the Geman and Reynolds' priors discussed in Chapter 3 have parameters tuned to recovering horizontal and vertical edges, or additionally in our extension, diagonal edges. In this section, we will extend the partitioning ideas to a diagonal case, and demonstrate the resulting scheme applied to a fourth test scene. Figure 5.9 shows this new test scene, which contains objects with horizontal, vertical and diagonal edges. Corrupting with uniform 3×3 blurring and $N(0,1)$ noise, the scene has energy 451.2 under a four neighbour Geman and Reynolds' prior with $\Delta=10$ and $d=3$.



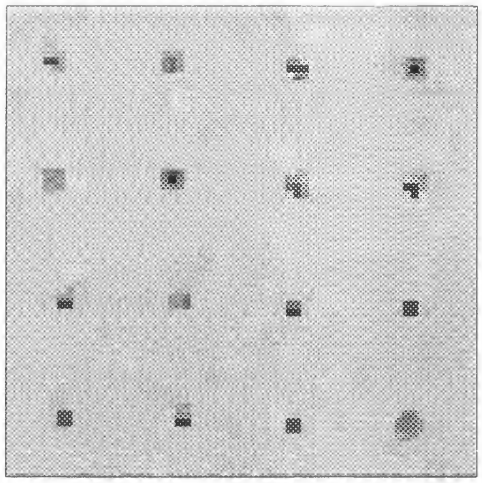
(a) (b)

Figure 5.9 (a) Fourth test image containing a mixture of blocks with horizontal-vertical edges and diagonal edges, and (b) record after 3×3 uniform blur and $N(0,1)$ noise.

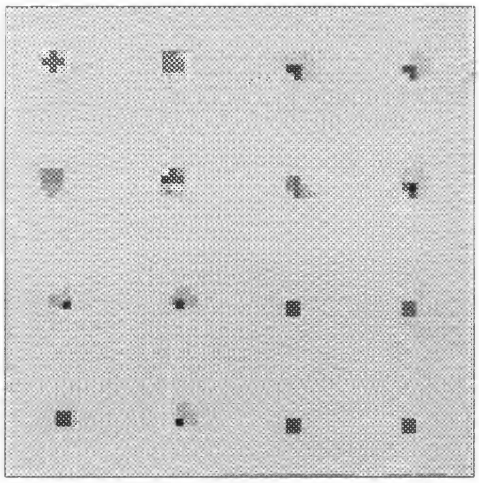
The partitioning ideas mainly carry over from the previous case considered. We will continue to use a disparity measure of the form given by Equation (5.2). The definition of Q_i^{ξ} retains the same form, however the definition of the blocks needs to be revised. Rows or columns of pixels are replaced by strips of pixels connected through their diagonally opposite corners, and running at 90° to each other across the pixel grid. Blocks are now defined to consist of a certain number of these tessellating parallel strips, the lengths of which will vary across the scene. It is possible to run all versions of the cascade discussed so far, standard, adaptive and window, using the diagonal partitioning scheme. For computational simplicity, we have continued to use a square window within which to apply the diagonal adaptive cascade. There are two points to note in this case, neither of which constitutes an objection to this cascade. Suppose we have a window of dimension $n \times n$ pixels, where n will be a power of 2 as a result of our construction. A horizontal partitioning line could be placed in $n-1$ possible sites; both directions of the diagonal partition have $2(n-1)$ possible sites (by considering the possible starting points of the partition along the edges of the window). If we continue to find the levels of the cascade by defining one pair of partitions, then three pairs of partitions, and so on, the diagonal cascade will have one more level than the corresponding horizontal-vertical cascade. Also by considering whether a pair of perpendicular diagonal partitioning lines intersect, we can see that it may no longer be the case that the cascade results in the sequence 4, 16, 64, ... regions.

Figure 5.10 shows six reconstructions of the latest test scene. All six reconstructions use ICM, and a four neighbour model with $\Delta=10$ and $d=3$. All the adaptive cascades use $\xi=2$. The reconstructions use (a) no cascade, (b) standard cascade, (c) a horizontal-vertical window cascade (after a horizontal-vertical adaptive cascade), (d) the previous reconstruction plus a diagonal window cascade, (e) a diagonal window cascade (after a diagonal adaptive cascade), and (f) the previous reconstruction plus a horizontal and vertical window cascade. A simulated annealing reconstruction was also obtained using a linear schedule from temperature 1 to 0 in 100 sweeps. This reconstruction is not shown, but had an energy of 462.5. The two best reconstructions both visually, and in terms of their energy, are those which use adaptive and window cascades incorporating both partitioning schemes, (d) and (f). The order of applying the windows with different partitioning schemes does not seem to have made a difference in this case ((d) and (f) are fairly comparable).

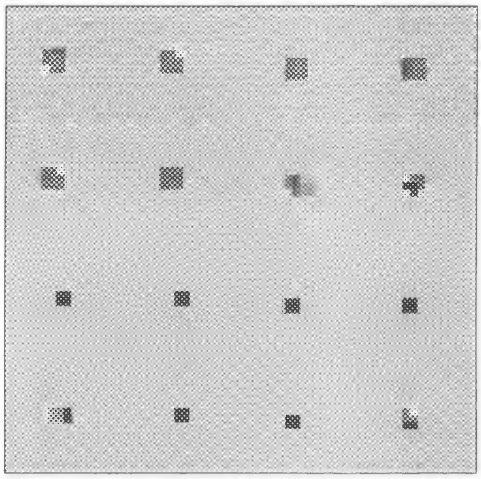
We will also present the results of various cascades with ICM on a more general first order scene, the image first used in Chapter 3, and shown in Figure 3.3(a). Figure 5.11 shows (a) the record after degradation by $N(0,4)$ noise and 3×3 blurring, and the restorations after (b) standard ICM, (c) a non-adaptive



(a) Standard ICM reconstruction
Energy = 504.2

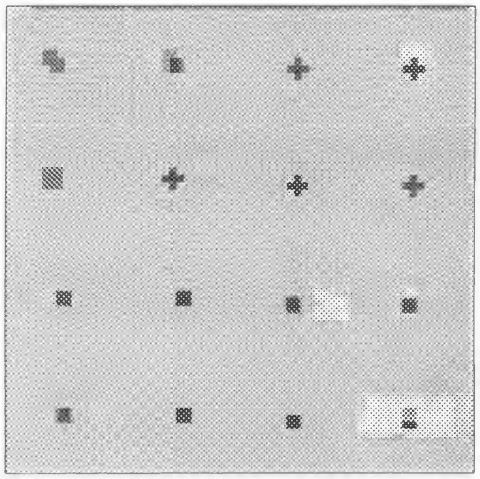


(b) Standard cascade reconstruction
Energy = 470.0

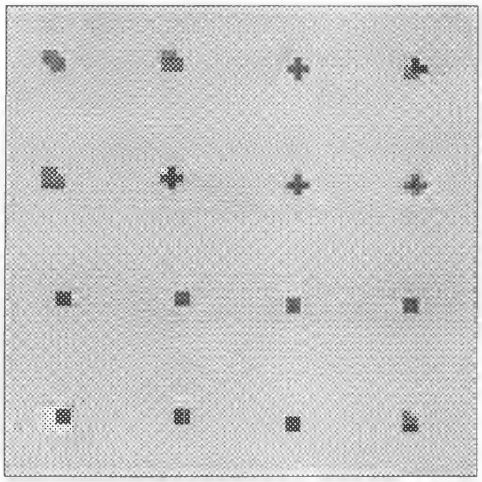


(c) Horizontal window cascade
Energy = 462.0

→

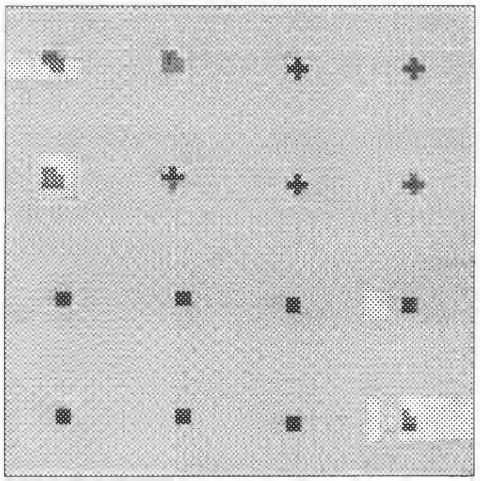


(d) (c)+diagonal window cascade
Energy = 452.8



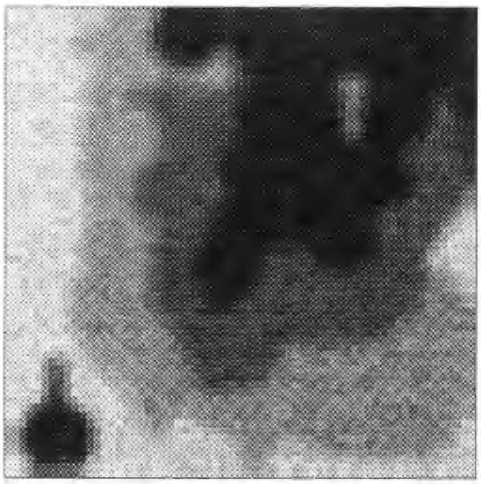
(e) Diagonal window cascade
Energy = 454.3

→

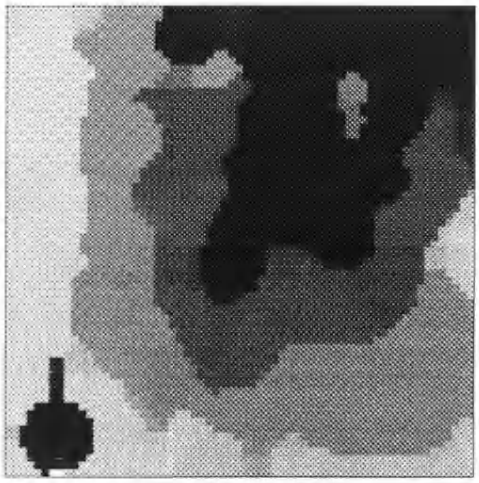


(f) (e)+horizontal window cascade
Energy = 452.3

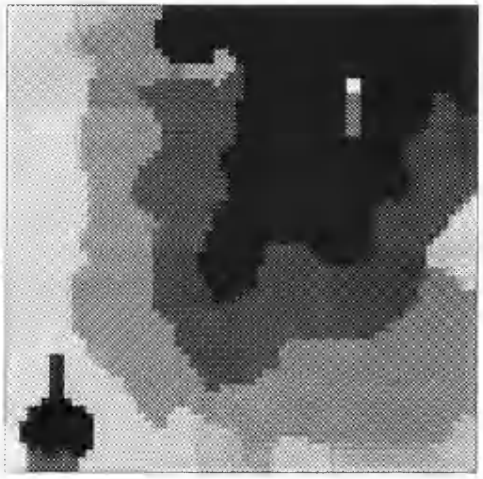
Figure 5.10 Six reconstructions of the fourth test scene, ξ=2.



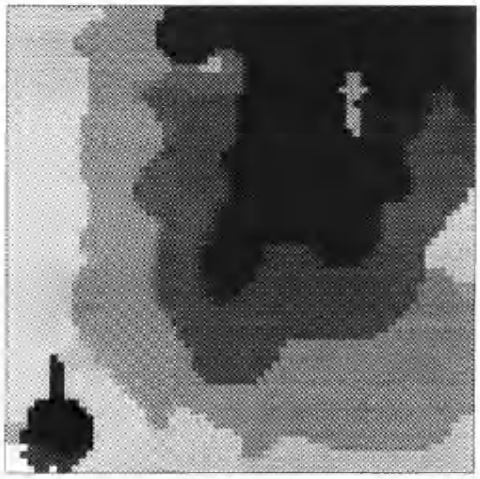
(a) Record after 3×3 uniform blur and $N(0,4)$ noise



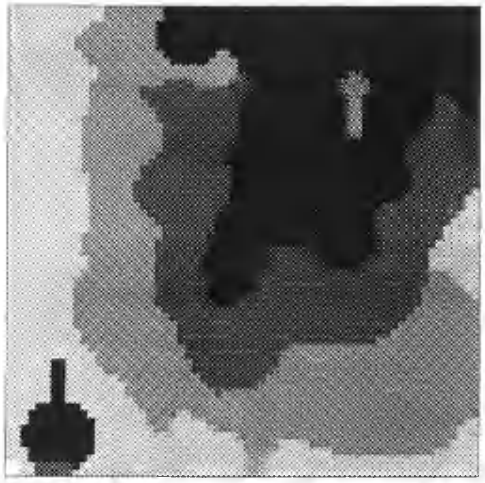
(b) Standard ICM reconstruction
Energy = 959.5



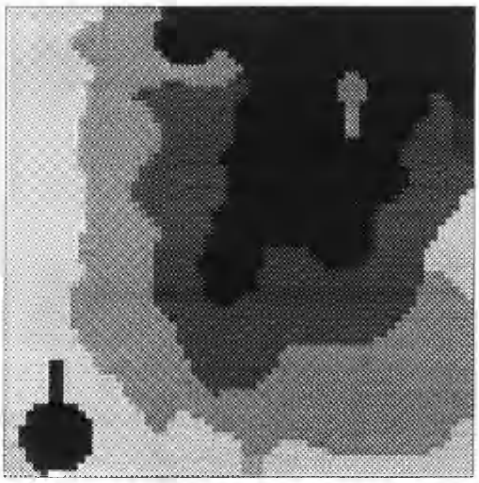
(c) Non-adaptive cascade
Energy = 973.6



(d) Adaptive cascade, ξ=4
Energy = 982.1



(e) (d)+horizontal window cascade
Energy = 916.0



(f) (e)+diagonal window cascade
Energy = 903.9

Figure 5.11 The record and various reconstructions for image shown in Figure 3.3(a).

cascade, (d) an adaptive cascade with ξ=4, (e) a series of horizontal-vertical window cascades with ξ=4, 2, 1 initialised with (d), and finally (f) a series of diagonal window cascades with ξ=4, 2, 1 initialised with (e). The model used is an eight neighbour Geman and Reynolds prior, with Δ=15 and d=3. As usual, a simulated annealing reconstruction was also obtained using a linear schedule from temperature 1 to 0 in 100 sweeps. This reconstruction is not shown, but had an energy of 916.9. The first point to note from (b), (c) and (d) is that the non-window cascades actually do worse than the standard ICM, with the adaptive cascade having a higher energy than the non-adaptive cascade. This suggests that any form of partition across the whole of the pixel grid is inadequate, as an examination of the original image might suggest. Reassuringly, the window cascades applied to the adaptive cascade reconstruction perform much better. Even with horizontal-vertical partitions alone, the reconstruction (e) attains the lowest energy so far. Comparing (b) and (f), the final cascade reconstruction has more clearly identified regions, both in terms of positioning the boundaries, and in terms of the grey-level variation within these boundaries (this point is less clear on the grey-level scale used for printing).

It seems that the modifications to the cascade algorithm we have considered in the chapter can be beneficial, at least in tackling the problems raised by our four test examples. In the next section, we will assess ICM cascade in application to reconstructing a very different type of image, and also in terms its computational expense compared to simulated annealing.

5.5 Some examples with ICM

So far in this chapter, we have mainly used simple test examples to demonstrate limitations of the cascade algorithm, which we have then attempted to tackle by extending the implementation. In this section, we will attempt to assess the performance of the revised cascade algorithm by reconstructing a 64×64 scene quite unlike the test examples used before. This scene is the digitised image of a face, it is almost certainly not first order in the sense described in Section 3.3, and its structure is far more complex than isolated foreground objects in uniform backgrounds. We hope that the partitioning scheme involved in the adaptive cascade will help to identify regions of the scene, and so give a visually good reconstruction. However, the primary motivation for cascade is to improve the performance of the minimisation algorithm with which it is used.

Four known degradations of the image are considered, these are the four combinations of a high or low noise, with high or no blurring. The four records are given in Figure 5.12(a)-(d), together with the details of the noise and blurring. The image has 64 grey-levels, and a Geman and Reynolds' prior is used with eight

neighbours, and parameters $\Delta=10$ and $d=5$. Notice that the parameter λ will differ for the four reconstructions since it is a function of the particular noise and blurring; the energies between records are not comparable.

We will use three reconstruction techniques. The first is standard single-site update ICM. The second is Gibbs sampler based simulated annealing, with a linear temperature schedule from 1 to 0 in 100 sweeps. The third is a combination of ICM cascades, starting with an adaptive cascade using horizontal-vertical partitions, and $\xi=4$. This is then followed by six window cascades with $\xi=4$, $\xi=2$ and $\xi=1$ for both horizontal-vertical, and then diagonal partitioning schemes. The four triplets of reconstructions are shown in Figures 5.13, 5.14, 5.15 and 5.16 as (a), (b) and (c) respectively. The energy of each reconstruction is quoted to one decimal place, and also the time taken as a multiple of the time taken for the corresponding ICM reconstruction, to the nearest integer. For the cascade reconstructions, this information is also broken down, and quoted to the same accuracy, for the individual cascade stages.

The first point to notice is that cascade achieves the lowest reconstruction energy for all four records. In two cases, it does this in approximately the same amount of time as simulated annealing which always achieves the second lowest energy, in one case it is quicker, and in the remaining case slower. When the breakdown of timings are considered for the different cascades, several patterns can be seen. Diagonal cascades take considerably longer than their horizontal-vertical counterparts. This is possibly due in part to the extra level (see Section 5.4.2), but mainly it is due to rather sub-optimal programming of the diagonal cascade. A more efficient implementation could probably reduce the computing time for the diagonal cascades to be comparable with the horizontal-vertical cascades. For all four records, the $\xi=4$ diagonal window cascade takes a large proportion of the total cascade time, but does not result in any reduction in energy. Generally the diagonal window cascades seem to give a lower return, in terms of energy reduction for time taken, than their horizontal-vertical counterparts. Even if no diagonal partitions had been considered, the energy after cascade would still have been lower than after simulated annealing in all four cases. In addition, the time taken would have been much reduced.

As regards the visual aspects of the reconstructions, the reconstructions from cascade do not always appear to be the best. Generally, cascade appears to divide the image into fewer, larger first order regions than the other two reconstruction techniques. In Figures 5.13 and 5.15, the low noise cases, this gives quite clean reconstructions which would provide good starting values for a higher order model. It is identifying large scale boundaries quite well, and avoiding some of the spurious regions found by standard ICM as a result of the higher order

(a) Record 1
Low noise $N(0,1)$
No blurring

(b) Record 2
High noise $N(0,16)$
No blurring

(c) Record 3
Low noise $N(0,1)$
Uniform 5×5 blurring

(d) Record 4
High noise $N(0,16)$
Uniform 5×5 blurring

Figure 5.12 Four records resulting from different corruptions of the face test image.

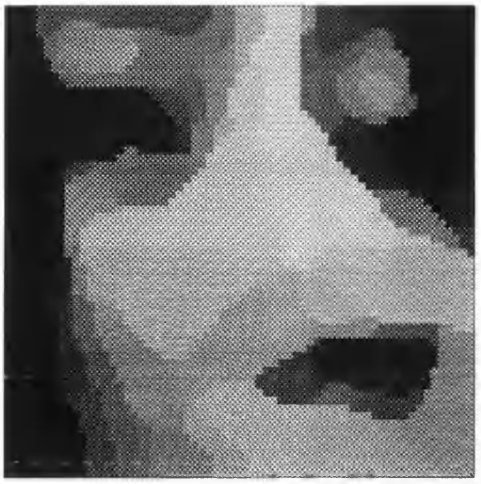
(a) ICM reconstruction
Energy = 1594.3
Time taken: 1 unit

(b) Simulated annealing reconstruction
Energy = 1524.8
Time taken: 36 units

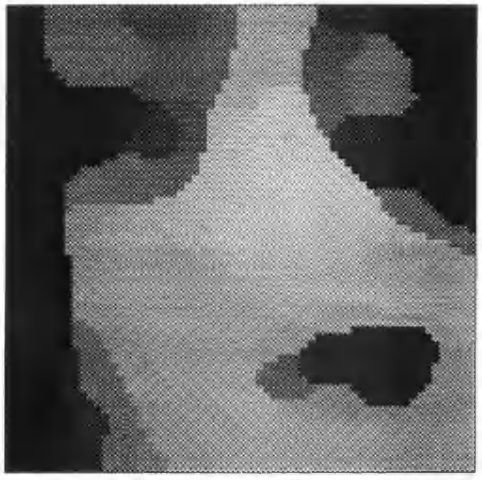
(c) Cascade reconstruction
Energy = 1483.4
Total time taken: 29 units

HV adaptive ξ=4: 1576.3 (1 unit)
HV window ξ=4: 1524.8 (2 units)
HV window ξ=2: 1503.7 (2 units)
HV window ξ=1: 1489.3 (2 units)
D window ξ=4: " (11 units)
D window ξ=2: 1489.1 (6 units)
D window ξ=1: 1483.4 (4 units)

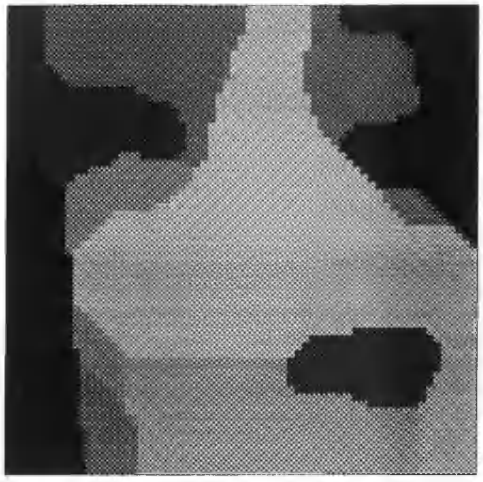
Figure 5.13 Three reconstructions of the first record (low noise, no blur).



(a) ICM reconstruction
Energy = 1300.4
Time taken: 1 unit



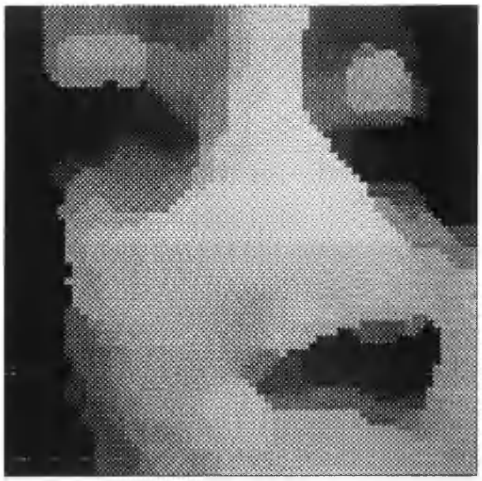
(b) Simulated annealing reconstruction
Energy = 1093.4
Time taken: 34 units



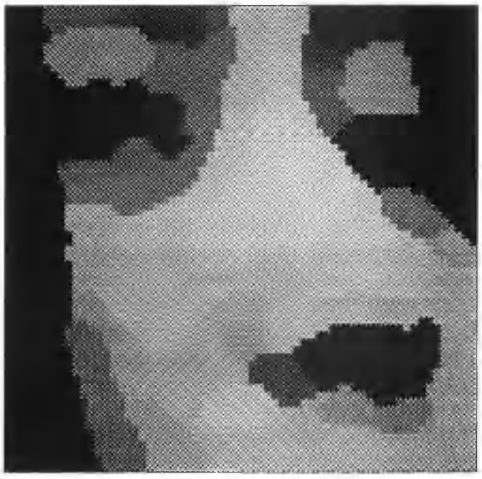
(c) Cascade reconstruction
Energy = 964.8
Total time taken: 20 units

HV adaptive ξ=4: 1062.9 (1 unit)
HV window ξ=4: 985.2 (1 unit)
HV window ξ=2: 974.6 (1 unit)
HV window ξ=1: 966.7 (2 units)
D window ξ=4: " (7 units)
D window ξ=2: 965.3 (5 units)
D window ξ=1: 964.8 (3 units)

Figure 5.14 Three reconstructions of the second record (high noise, no blur).



(a) ICM reconstruction
Energy = 1331.7
Time taken: 1 unit



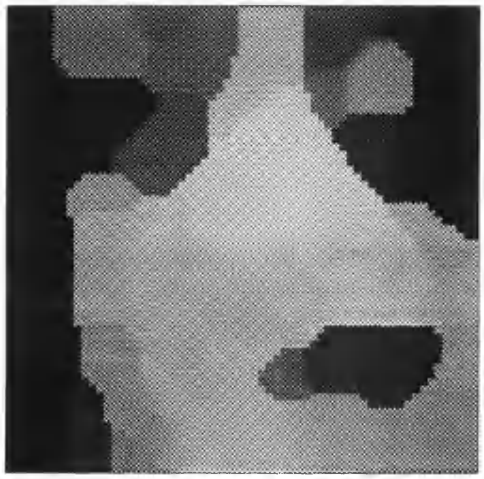
(b) Simulated annealing reconstruction
Energy = 1137.4
Time taken: 12 units

(c) Cascade reconstruction
Energy = 1105.5
Total time taken: 18 units

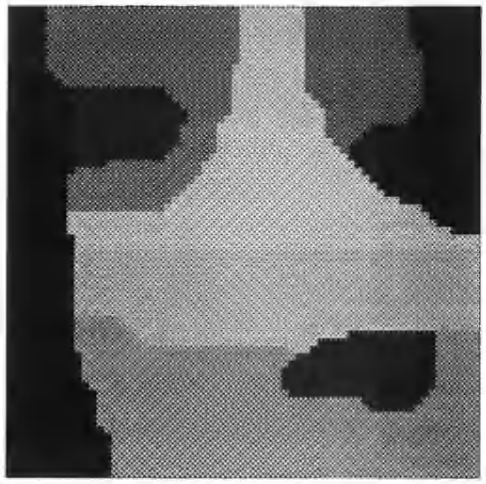
HV adaptive ξ=4: 1281.4 (1 unit)
HV window ξ=4: 1218.4 (2 units)
HV window ξ=2: 1145.9 (1 unit)
HV window ξ=1: 1115.6 (1 unit)
D window ξ=4: " (6 units)
D window ξ=2: 1114.7 (4 units)
D window ξ=1: 1105.5 (2 units)

Figure 5.15 Three reconstructions of the third record (low noise, high blur).

(a) ICM reconstruction
Energy = 1162.3
Time taken: 1 unit



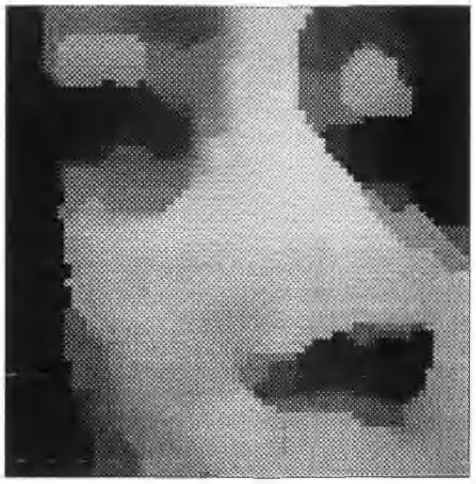
(b) Simulated annealing reconstruction
Energy = 955.1
Time taken: 12 units



(c) Cascade reconstruction
Energy = 844.8
Total time taken: 10 units

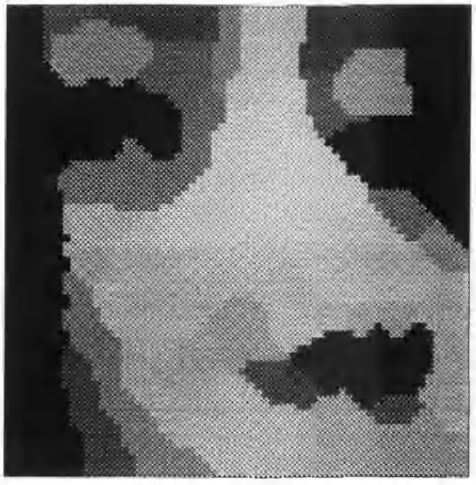
HV adaptive ξ=4: 888.2 (1 unit)
HV window ξ=4: 859.0 (1 unit)
HV window ξ=2: 845.9 (1 unit)
HV window ξ=1: 845.0 (1 unit)
D window ξ=4: " (4 units)
D window ξ=2: 845.0 (3 units)
D window ξ=1: 844.8 (1 unit)

Figure 5.16 Three reconstructions of the fourth record (high noise, high blur).



(a) Reconstruction from first order ICM reconstruction (Figure 5.15(a))
Energy = 8926.5

(b) Reconstruction from first order simulated annealing reconstruction (Figure 5.15(b))
Energy = 8660.2



(c) Reconstruction from first order cascade reconstruction (Figure 5.15(c))
Energy = 8474.3

Figure 5.17 Second order ICM reconstructions of the third record (low noise, high blur).

nature of the underlying image. As a point of interest, Figure 5.17 shows the three second order ICM reconstructions initialised from the three first order reconstructions of the third record, shown in Figure 5.15. The model used is a second order Geman and Reynolds' prior with eight neighbours, Δ=5, and d=5. The ordering of the image energies remain the same, with the lowest second order energy given by the reconstruction initialised by the cascade first order reconstruction.

In the remaining two cases, the high noise Figures 5.14 and 5.16, the reconstructions possibly appear over-smooth, exhibiting less structure which, to the human eye, would identify a face. These reconstructions are still, however, low energy realisations; this is perhaps an example of the problems of MAP estimation.

Generally, the conclusion of this chapter must be that this modified cascade does appear to be improving the performance of the ICM algorithm. Unfortunately, the particular partitioning scheme can be crucial to the improvements, at least in conjunction with ICM, and more work is needed to identify suitable schemes. In order to gain more understanding of the reasons why the improvements might be occurring, we need to investigate the effects of the cascade algorithm further. This will be the topic of the Chapter 6.

Chapter 6: Simulated Annealing and the Cascade Algorithm

6.1 Introduction

6.1.1 Outline of chapter

In Section 4.1.1, the need for multiple site update methods was discussed. One of the reasons given there is that it can be prohibitively slow to move around the complex energy function by updating a single pixel at a time. Employing the redefinition of the cascade algorithm given in Chapter 5, it can be seen that higher levels of the cascade correspond to taking large steps around some subset of the energy function, determined by the blocking scheme. As the last chapter demonstrated, this appears to improve the performance of ICM in reconstructing certain types of image. One speculative analysis of cascade is that it is allowing us to avoid certain high energy areas, more rapidly reaching better regions of the energy function. Having reached such a region, the next level of the cascade then allows us to repeat this procedure at a finer level of resolution. Working with ICM, it is difficult to assess this argument since the algorithm is a strictly downhill search procedure, and there is no real exploration of these regions. In this chapter, we will consider cascade in conjunction with simulated annealing.

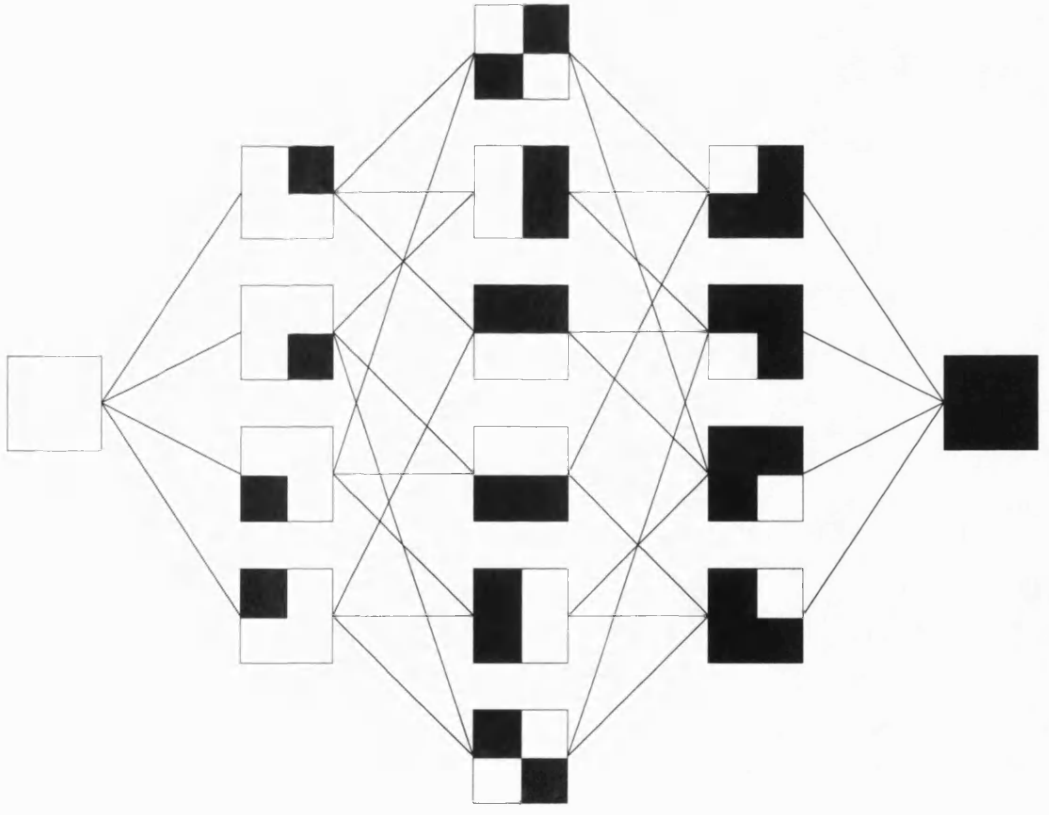


Figure 6.1 The image graph formed by the configurations of a 2×2 binary image.

Stochastic techniques such as simulated annealing are required in the image problem because of the high dimensionality of the energy function. This dimension difficulty also confronts us when we try to investigate cascade. Consider Figure 6.1 which depicts the sixteen possible configurations of a 2×2 binary image, one of the simplest possible images. Associated with each configuration is the corresponding energy value of that image. For our purposes, this energy function could be visualised as a 3-dimensional surface, out of the page and above the configurations. The lines in Figure 6.1 denote transitions that may be made between two images by changing a single pixel value. Single-pixel update methods may move around the energy surface by following the paths defined between images by these lines.

The implementation of any form of cascade would not be appropriate for an image as small as that in Figure 6.1. However, the underlying idea is still clear. In the early stages of the cascade algorithm, certain images, or nodes in this undirected graph, would be deleted, together with their associated energy values. New paths would then be created between the pairs of images which could be interchanged by a single update of a cascade block of pixels. For example, suppose we were to constrain the left and right halves of our 2×2 binary image to be cascade pixel blocks. This would result in the reduced image graph shown in Figure 6.2. The energy of these four images remains the same, but the topology of the energy surface has changed.

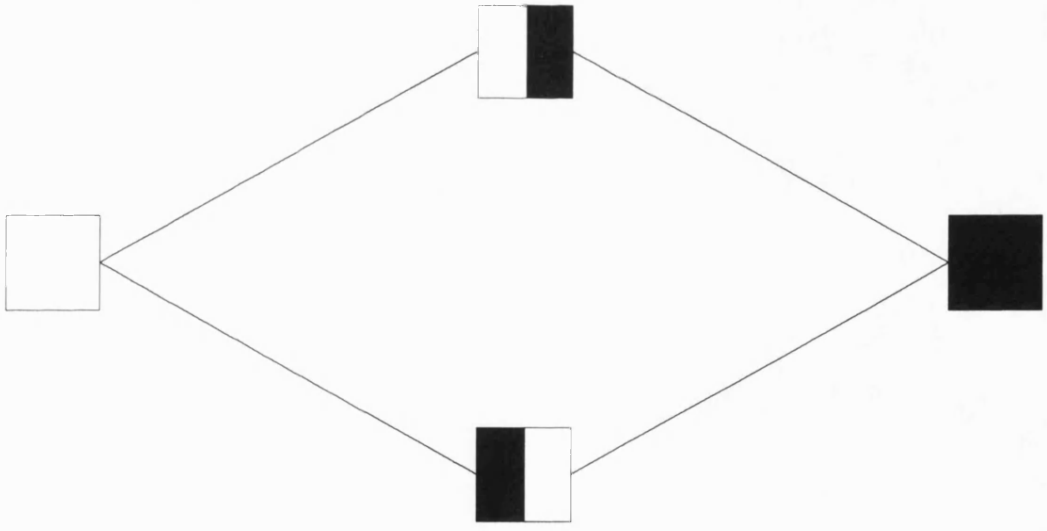


Figure 6.2 An image graph resulting from a possible cascade constraint.

On a grid of $|S^0|$ pixels, each of which can take N values, each image has $|S^0|(N-1)$ neighbours in the graph (here the term neighbour is used to indicate that a transition line exists between the images, and not in any Markov random field sense). Any increase in the dimensions of the grid, or the number of pixel

colourings, increases the number of neighbours for each image. This increase would complicate the graph to the extent that it would become difficult to represent graphically. Ideally, in order to investigate cascade, it is these larger problems which need to be considered. However, an investigation of such large problems seems unmanageable at present. Instead, we intend to consider experiments on a function defined on a regular $n \times n$ lattice. Wherever possible, these experiments will be constructed to retain features of the image problem, in the hope that any conclusions we may be able to reach will then also apply to the image problem.

We will begin by stating the alternative minimisation problem, and defining a cascade equivalent in this situation. The differences in structure between the $n \times n$ lattice and the image graph require slight modifications to be made to the particular Hastings algorithms usually applied. These modifications are described in Section 6.2, together with an incidental comment which arises regarding a computational saving for the Gibbs sampler considered by Geman & Reynolds (1992). The use of our small alternative problem allows us to monitor the performance of simulated annealing, in particular the convergence under the theoretical conditions described in Chapter 2. As a result, various aspects of simulated annealing without cascade are considered in Section 6.3; the introduction of cascade to the system is deferred until Section 6.4. Finally, in Section 6.5, we return to images, and attempt to use some of the insight gained from the lattice problem in applying simulated annealing with cascade to image reconstruction.

6.1.2 An alternative minimisation problem with cascade

We will consider a regular $n \times n$ lattice which has undirected edges between each node and its four nearest neighbours (or those nearest neighbours which exist at the edges of the lattice). Denote the nodes of the lattice by $\{\mathbf{x}\}$. In drawing our analogy with the image problem, the nodes could be considered to represent different images. The existence of an edge between two nodes is taken to indicate that a transition from one image to the other could be accomplished by a single pixel update. We will then define some function $V(\mathbf{x})$ on the lattice, with the minimum and maximum of $V(\,)$ on the lattice being 0 and 1 respectively. Continuing the image analogy, this function $V(\,)$ will be taken to be equivalent to the image energy function. The lattice problem is to find the minimising node $\mathbf{x}_{\min}$, where $V(\mathbf{x}_{\min})=0$.

In order to define some equivalent of a cascade algorithm on the lattice, it was decided to adopt a deletion of every second node in both directions, resulting in a reduced $n/2 \times n/2$ sublattice. Edges exist in the sublattice in the same way as in the full lattice; each surviving node is connected to the four nearest neighbour surviving nodes, except at edges where there are fewer connections. This scheme is illustrated in Figure 6.3; (a) represents the full lattice with nodes denoted by $\circ$,

in (b) those nodes which are to be retained at the first level of the cascade are denoted by •. Except at edges, each ∘ is connected to its four nearest neighbour ∘, each • is connected to its four nearest neighbour •. Smaller sublattices, equivalent to higher cascade levels, could be generated by the same deletion and connection scheme applied to the • lattice.

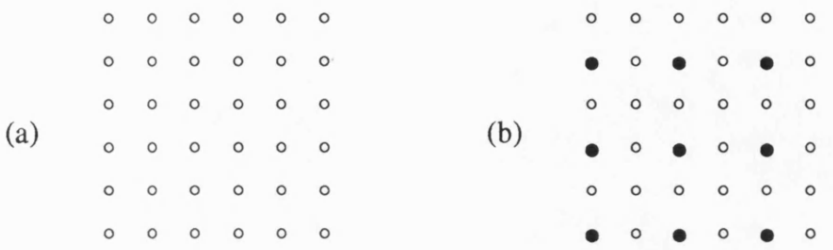


Figure 6.3 Formation of the sublattice.

We will consider the suitability of our lattice cascade. In the node deletion scheme's favour, it does have the key feature of a systematic thinning, with reconnection between surviving nodes which were closest in the original lattice. However, it does not correspond directly to the reduction occurring in the image graph when cascade is applied in the image problem. Under the original implementation of the cascade algorithm, the cascade blocks were predetermined $2^m \times 2^m$ groups of pixels. This blocking scheme was independent of the data, in much the same way that this node deletion scheme is independent of the function $V(\ )$. However, the ICM examples in Section 5.3 demonstrated that this was not the optimal policy, and better results were obtained by using a data-driven partitioning scheme. By selecting a partition, we are determining the reduced image graph. If the blocks are relatively well chosen for the scene, then this graph should contain images where many of the block values are close to their minimising values, and these realisations may have fairly low energy. However, since the blocks are not constrained in their values, the reduced graph must then also contain images where many of the blocks values are far from optimal, and these may have high energy. It is possible that a well chosen partition may result in a more easily minimised energy function, so it is not surprising that the adaptive schemes can perform better. We have lost that advantage in the node deletion.

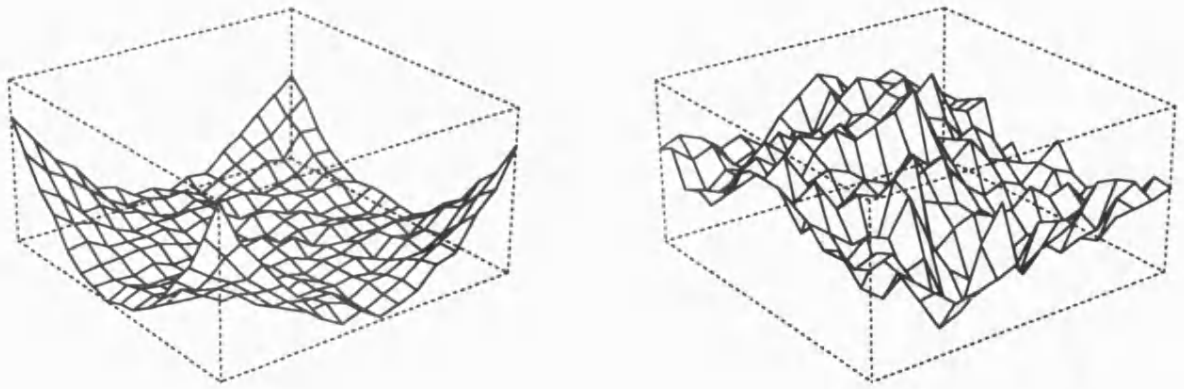
In terms of the relative smoothing effects of the node deletion and the cascade graph deletion, a comparison is difficult because we can say so little about the energy on the image graph. However, if we were to use a 1-dimensional analogy for the image system, for example a single row of the lattice with only left and right connections, then the corresponding $V(\ )$ on the nodes would have the

appearance of a skyline for a mountain range. Deleting every other node could only increase the smoothness of the skyline, reducing the height of "mountains" and the depth of "valleys". The climb required to reach any retained node from another retained node is either reduced, or remains the same. On the image graph, it may not be the case that there is necessarily a similar increase in smoothness as a result of the graph reduction. Considering Figure 6.3(b), we can see that we will lose certain routes which existed between • nodes in the undeleted lattice. It is possible that the energy climb required to reach some • from any other • may actually be increased by the node deletion. By using our 2-dimensional lattice, we have at least avoided this 1-dimensional problem.

The two test $V(\ )$, the functions to be minimised over the lattice, are shown in Figure 6.4. The full lattices contain 16×16 nodes; the figure also illustrates the reduction in these $V(\ )$ after the deletion procedure producing an 8×8 and a 4×4 sublattice. The first test function, shown in (a), was generated by adding a small amount of noise to a quadratic surface. The effect of this noise is minimal where the function is steeply decreasing, but in the centre of the lattice, where the quadratic is minimised, the noise leads to a large number of local minima, all of about the same size. A deterministic, downhill algorithm would probably be able to locate the central low basin, but would inevitably become trapped in one of the local minima. The reduced functions exhibit similar, though less extreme, behaviour. This test function (a) will be identified by the label smooth.

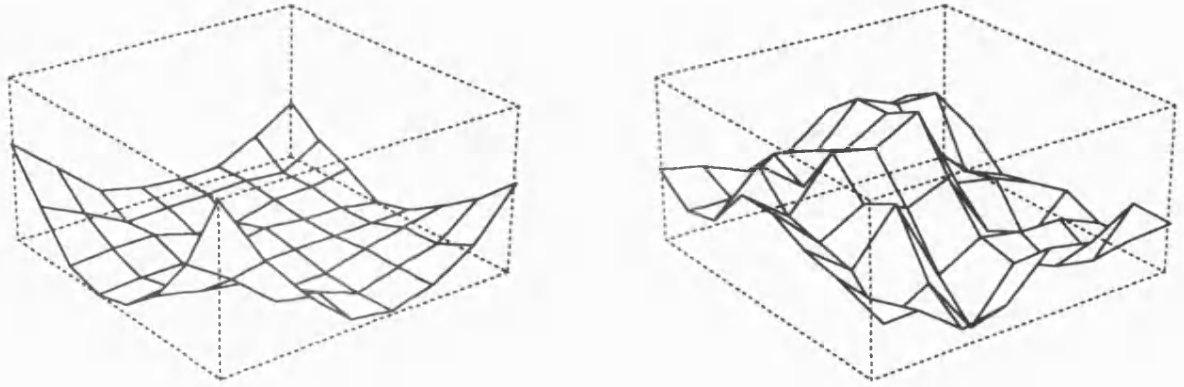
The second test function, shown in (b), was generated by hand. It contains several distinct local minima, well separated by a ridge of higher values. A downhill search procedure would certainly have difficulty in finding the global minimum before becoming trapped in some local minimum. This test function will be identified by the label rough. Neither test function contains the global minimiser in any deletion sublattices, and in both cases the minimisers on different lattice levels are not necessarily close on the full lattice.

Our justification for these two test functions is that they have characteristics which we believe might also be found in the image energy function. We might expect regions of unlikely, high-energy scenes where most of the pixels take poor values, and almost any update will decrease the energy. This may be particularly pronounced on cascade-reduced graphs where, as mentioned before, large numbers of pixels all do well, or all do badly. In low energy regions, where the images differ at only a few critical pixels, many local minima may exist, and there may be less consistent gradient to the function. We would like to view (b) as some low energy region of the function. With a change of scale, the cascade-reduced view (f) of this region could be seen as some small central area of (a), itself a cascade-reduced view of the whole of the energy function.



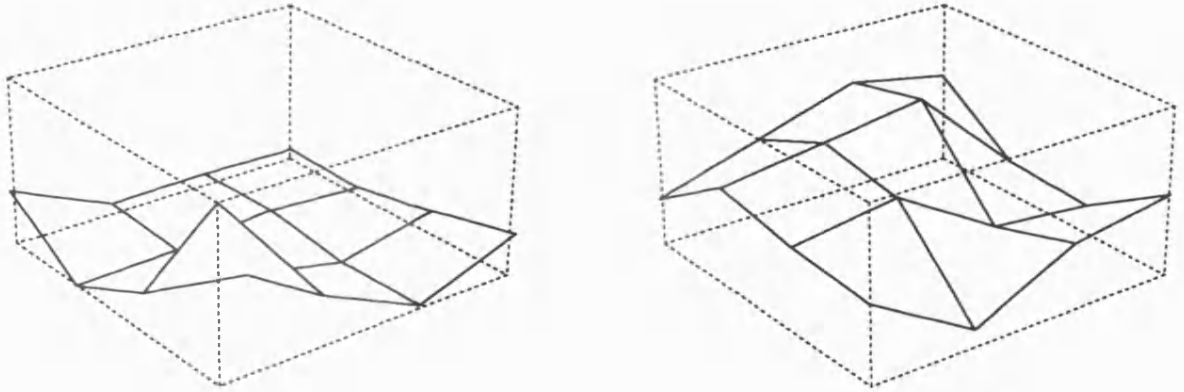
(a) ↓ (b) ↓

Function values over the full 16×16 lattice for two test examples



(c) ↓ (d) ↓

Function values over the reduced 8×8 lattice for the same test examples



(e) (f)

Function values over the reduced 4×4 lattice for the same test examples

Figure 6.4 Three layers of deletion for the two test functions.

6.2 Hastings algorithms on the lattice

6.2.1 The modified Metropolis algorithm and Gibbs sampler

In the image case, the probability density function from which we wish to sample, is given by the renormalised exponential of the negative energy function, possibly at some temperature T_k (Equation (2.9)). Since on the lattice, we wish to regard our test functions $V(\ )$ as some form of energy function, we will define the temperature T_k density on the lattice by

$$p_k(\mathbf{x}) = \frac{\exp(-V(\mathbf{x})/T_k)}{\sum_{\{\mathbf{x}'\}} \exp(-V(\mathbf{x}')/T_k)}. \tag{6.1}$$

High energy nodes have low probability, low energy nodes have high probability. As the temperature T_k becomes smaller, the density is increasingly concentrated on the minimum energy node, $\mathbf{x}_{\min}$.

The general Hastings algorithm, together with two examples, the Gibbs sampler and the Metropolis algorithm, was described in Section 2.2. The algorithm attempts to draw a sample from the target distribution by following an iterative procedure. Starting at some state $\mathbf{x}$, a new realisation $\mathbf{x}'$ is proposed according to the distribution $q(\mathbf{x},\mathbf{x}')$. The proposal is then either accepted, with probability $\alpha(\mathbf{x},\mathbf{x}')$, or rejected, in which case we retain $\mathbf{x}$. The sampling distribution generated by the algorithm will converge in distribution to the target distribution, provided that the associated Markov chain is aperiodic, irreducible and satisfies detailed balance (Equation (2.12)). The different members of the Hastings family of algorithms can be characterised by their proposal distributions; the forms of $q(\ )$ and $\alpha(\ )$ for the Metropolis algorithm and the Gibbs sampler, as applied to images, are given in Equations (2.15)-(2.18). In this section, our intention is to identify the Hastings algorithms on the lattice which correspond in spirit to these two specific examples.

In single-site updating for images, the proposal for a new realisation is drawn from those images which could be obtained from the current realisation by updating a single pixel. In our lattice analogy, the equivalent move from any node would be to one of its nearest neighbour nodes. In block updating, the possible image proposals involve the updating of a single cascade block; our lattice equivalent is a move between neighbouring sublattice nodes.

We will first consider the Gibbs sampler equivalent on the lattice. The essence of the Gibbs sampler is that realisations will be proposed in proportion to their probability under the target distribution, so we are more likely to propose a low energy, rather than a high energy, realisation. We will use the suffix n to indicate that the corresponding quantity is defined for the $n\times n$ lattice. So, to sample from $p_k(\ )$ given by Equation (6.1) on the lattice, we will use the proposal

$$q_n(\mathbf{x},\mathbf{x}') = \frac{p_k(\mathbf{x}')}{\sum_{\mathbf{x}''\in\tau_{\mathbf{x}}} p_k(\mathbf{x}'')}, \qquad \mathbf{x}'\in\tau_{\mathbf{x}} \tag{6.2}$$

where $\tau_{\mathbf{x}}$ is the set of neighbouring nodes of $\mathbf{x}$, taken to include $\mathbf{x}$ in line with the original Gibbs sampler. In order to satisfy detailed balance, Equation (2.13) determines the acceptance probabilities to be of the form,

$$\alpha_n(\mathbf{x},\mathbf{x}') = \min\left(1\ ,\ \frac{\sum_{\mathbf{x}''\in\tau_{\mathbf{x}}} p_k(\mathbf{x}'')}{\sum_{\mathbf{x}'''\in\tau_{\mathbf{x}'}} p_k(\mathbf{x}''')}\right), \qquad \mathbf{x}'\in\tau_{\mathbf{x}}. \tag{6.3}$$

In the image case, the Gibbs sampler always had zero rejection. This resulted from the fact that the set of proposals, in that case, was the set of the N possible colourings for the pixel being updated. Independently of the particular realisations $\mathbf{x}$ and $\mathbf{x}'$ differing at just this one pixel, this set contained the same images. However, since the summations in the numerator and denominator in this case are over different sets, we no longer always have zero rejection.

We will now consider the lattice equivalent of the Metropolis algorithm as applied to images. The spirit there was to propose the new pixel value uniformly from among the $N-1$ alternatives to the current value. Our alternatives on the lattice are the neighbouring nodes, so

$$q_n(\mathbf{x},\mathbf{x}') = \frac{1}{|\tau_{\mathbf{x}}|-1}, \qquad \mathbf{x}'\in\tau_{\mathbf{x}},\ \mathbf{x}'\neq\mathbf{x} \tag{6.4}$$

$$\alpha_n(\mathbf{x},\mathbf{x}') = \min\left(1\ ,\ \frac{(|\tau_{\mathbf{x}}|-1)p_k(\mathbf{x}')}{(|\tau_{\mathbf{x}'}|-1)p_k(\mathbf{x})}\right), \quad \mathbf{x}'\in\tau_{\mathbf{x}},\ \mathbf{x}'\neq\mathbf{x}. \tag{6.5}$$

On the lattice, not every node has the same number of neighbours. This leads to a non-symmetric proposal, which, strictly speaking, means that the algorithm should not be called a Metropolis algorithm.

6.2.2 Application of Hajek's result on the lattice

In Section 2.3, we discussed two theoretical results concerning simulated annealing. One result was based on the Gibbs sampler, and the other, stronger result due to Hajek (1988), was based on a Hastings algorithm with a symmetric proposal distribution, such as the Metropolis algorithm. Although in this section, we have been describing fixed temperature sampling, in the next section, we will wish to consider simulated annealing. It might be of great benefit there, to be able to apply Hajek's result, the conditions of which are given in Equation (2.20).

In our current Metropolis equivalent on the lattice, the proposal distribution, given in Equation (6.4), is non-symmetric when either of the nodes involved lies at the edge of the lattice. In order to apply Hajek's result, we wish to make a

modification to the problem which effectively treats all nodes as internal. To do this, we will introduce an additional set of lattice nodes, $\{\tilde{\mathbf{x}}\}$ say, one for each missing neighbour of every edge node, with each $\tilde{\mathbf{x}}$ connected only to the relevant edge node. The number of neighbours of any $\mathbf{x}$ is then equal to four. Next, we will extend the function $V(\,)$ to take the value $+\infty$ on all $\tilde{\mathbf{x}}$, so that $p_k(\tilde{\mathbf{x}})=0$, $\forall$ additional $\tilde{\mathbf{x}}$. This modification will not affect proposals between internal nodes; however at an edge site, any neighbour, including the relevant $\tilde{\mathbf{x}}$, will be proposed with probability ¼. In the event that an $\tilde{\mathbf{x}}$ is proposed, the move will always be rejected because the acceptance probability, given by Equation (6.5), is proportional to the probability of the proposed node, which in this case will be zero. Using this modification, we retain a valid Metropolis-Hastings algorithm satisfying detailed balance, but we also have a symmetric proposal which permits the application of Hajek's result.

6.2.3 A comment on the Geman and Reynolds' truncated algorithm

In the course of the work in Section 6.2.1, we noticed an incidental connection with a suggestion made in Geman & Reynolds (1992). This paper recommends a slight adaptation of the Gibbs sampler, which they call the truncated algorithm, aimed at reducing the computational burden of simulated annealing for grey-level images. They suggest that when updating any pixel s, rather than considering all N possible grey-levels, the choices are truncated to those levels lying within some prespecified distance $r>0$ either of the current grey-level, or of its record, or of any of its prior Markov random field neighbours. Levels satisfying this criterion define a new set, $\tau_{\mathbf{x}}^s$, of neighbouring images of the image $\mathbf{x}$. The next realisation $\mathbf{x}'$ is generated by sampling from $\tau_{\mathbf{x}}^s$ with probabilities proportional to the relative probabilities of the members of $\tau_{\mathbf{x}}^s$,

$$P(\mathbf{x},\mathbf{x}') = \frac{p(\mathbf{x}')}{\sum_{\mathbf{x}''\in\tau_{\mathbf{x}}^s} p(\mathbf{x}'')}, \qquad \mathbf{x}'\in\tau_{\mathbf{x}}^s. \tag{6.6}$$

The computational savings come in the calculation of the normalising constant since $|\tau_{\mathbf{x}}^s|$ may be considerably smaller than N. Geman and Reynolds claim that the probability of selecting an image not in $\tau_{\mathbf{x}}^s$ would be negligible under the standard Gibbs sampler, and so the outcome will not be greatly affected.

Suppose we consider any image $\mathbf{x}'$ in $\tau_{\mathbf{x}}^s$. For certain choices of $\mathbf{x}'$ and the truncation parameter r, the two sets $\tau_{\mathbf{x}}^s$ and $\tau_{\mathbf{x}'}^s$ may not have exactly the same members. This situation is much the same as was considered for the lattice Gibbs sampler in the last section. Regarding Equation (6.6) as the proposal distribution for a Hastings algorithm, the corresponding $\alpha(\mathbf{x},\mathbf{x}')$ would be as given by Equation (6.3). This does not necessarily give zero rejection, and so it can be seen

that this truncated algorithm does not satisfy detailed balance. Although adopting the possibility of rejection, in the form of Equation (6.3), would rectify this problem, the computational savings originally intended might well now be outweighed by the calculation of $\alpha(\mathbf{x},\mathbf{x}')$.

Geman and Reynolds do not explicitly identify this problem; they implement the truncated Gibbs sampler in the form that it is has been described. However, they briefly state that a modification of the truncation procedure, effectively the proposal distribution from a Hastings point of view, is possible. They suggest that this modification associates the set $\tau_{\mathbf{x}}^{s}$ with local sections in a slightly restricted image space, and allows the usual results on simulation and annealing to be applied. Stander (1992) has carried out a simulation study which suggests that the Geman and Reynolds' implementation compares well with the standard Gibbs sampler, and does represent a considerable computational saving.

Generally, the Gibbs sampler would be used in preference to the Metropolis algorithm for grey-level images, because a uniform proposal from among the N choices often appears to waste the chance of an update by proposing unlikely grey-levels. One attractive alternative to the truncated Gibbs sampler might be to consider the same approach, but now applied to a Metropolis-Hastings algorithm. At pixel s, proposals would only be drawn from the restricted set of neighbours, $\tau_{\mathbf{x}}^{s}$, and the forms of $q(\,)$ and $\alpha(\,)$ would be as given by Equations (6.4) and (6.5). The computational cost involved in this appears greater than that of the standard Metropolis algorithm, since membership of $\tau_{\mathbf{x}}^{s}$ must be calculated, as well as the additional factor in $\alpha(\,)$. However, this truncated algorithm may have an overall gain in efficiency, since it is preselecting images with a higher chance of acceptance. It can be seen that irreducibility is not lost by either of these truncated schemes, provided $r \neq 0$. This involves noting that the value of $\mathbf{X}_s$ can always move by one grey-level at each update, and so can attain the whole range $\{0, \ldots, N-1\}$, whatever values the rest of the image takes.

6.2.4 Monitoring the sampling distribution

When we use any Hastings algorithm to sample either at fixed temperature, or at decreasing temperatures in the case of simulated annealing, the hope is that at each step, we are generating a realisation from a distribution close to the appropriate target distribution. In the imaging context, we monitor the sequence of realised images, but not the sequence of distributions from which we are sampling, for the obvious reasons of dimension. However, on reasonably sized lattices, it is computationally feasible to calculate all the transitions $P^{k}(\mathbf{x},\mathbf{x}')$, the probability of being at node $\mathbf{x}'$ at step $k+1$, given that at step k we were at node $\mathbf{x}$. Given the initial sampling distribution, $\pi^{(0)}(\mathbf{x})$, this allows us to calculate the sampling distribution $\pi^{(k)}(\mathbf{x})$ at any positive step k, using the recursion

$$\begin{aligned}\pi^{(k+1)}(\mathbf{x}) &= P(\text{ at node } \mathbf{x} \text{ at step } k+1 \text{ }) \\ &= \sum_{\mathbf{x}'} P(\text{at node } \mathbf{x}' \text{ at step } k)\ P(\text{move from } \mathbf{x}' \text{ at } k \text{ to } \mathbf{x} \text{ at } k+1) \\ &= \sum_{\mathbf{x}'} \pi^k(\mathbf{x}')\ P^k(\ \mathbf{x}', \mathbf{x}\ ). \end{aligned} \tag{6.7}$$

In the fixed temperature case, the transitions $P^k(\mathbf{x}',\mathbf{x})$ are independent of k. However in simulated annealing, the distribution $p_k(\ )$ defined in Equation (6.1) is changing at each step, and the transition probabilities will be a function of this distribution through $q_n(\ )$ and $\alpha_n(\ )$.

In addition to calculating the sampling distribution at each step, from this information it is also possible to calculate various quantities of interest. In the fixed temperature case, we might want to monitor some distance measure between the sampling distribution and the target distribution, in an attempt to assess convergence. In the simulated annealing case, which we will consider in the next two sections, we are more interested in assessing how well the minimisation is progressing. We have chosen to track the expected energy of the distribution, and the probability of sampling $\mathbf{x}_{\min}$. Ideally, we would like the expected energy to be low, and the probability of selecting $\mathbf{x}_{\min}$ to be high. However, since in imaging we draw a single realisation from the sampling distribution at each step, it is debatable which diagnostic is more important. If we have a high probability of selecting the global minimiser, but also a relatively high expected energy, then if we fail to select the global minimiser, we may otherwise do quite badly. On the other hand, if both the expected energy and the probability of selecting $\mathbf{x}_{\min}$ are low, then we may have little chance of actually finding the global minimum, but we are fairly certain of at least a reasonably low energy estimate. On balance, the expected energy may be a better indicator of the minimisation performance, since in the image case, we can only realistically hope to find a good local minimum.

It is common practise to present the lowest energy image obtained at any sweep of simulated annealing as the final reconstruction. As a result, we have also chosen to monitor the probability of having been at $\mathbf{x}_{\min}$ at any step. This statistic requires a slight modification of the calculation given in Equation (6.7). Rather than monitoring the probabilities of being at the n^2 nodes at step k, we monitor the $2n^2$ conditional probabilities of being at node $\mathbf{x}$ at step k given whether or not we have visited $\mathbf{x}_{\min}$ by step k. The probability of having been at the global minimiser by some step k is then the total over the nodes of the conditional probabilities of being at any particular node having sometime visited $\mathbf{x}_{\min}$. The probability of being at a particular node is the sum of the two conditional probabilities for that node; the path either has, or has not, visited $\mathbf{x}_{\min}$ by that step.

6.3 Annealing on the lattice

6.3.1 Hajek's result and the logarithmic schedule

We are now in a situation where we have defined an alternative problem to be investigated, and where we have outlined the aspects of the problem in which we will be interested. In this section, we will monitor the sampling distribution on the lattice under various simulated annealing schedules not incorporating cascade, in an attempt to assess how well the procedure performs.

In Section 2.3.1, we discussed temperature schedules for simulated annealing asymptotically to find the global minimum of the energy function. The stronger of the two results discussed there is Hajek's theorem, Equation (2.20), which applies to simulated annealing based on the Metropolis algorithm. This theorem states that we have the desired convergence of the sampling distribution to the target distribution, if, and only if, the schedule T_1, T_2, ... is a sequence of positive numbers with $T_1 \geq T_2 \geq \ldots$, $\lim_{k\to\infty} T_k = 0$ and

$$\sum_{k=1}^{\infty} \exp(-d^* / T_k) = +\infty$$

$$\text{where} \quad d^* = \max_{\text{local minima } \mathbf{x}} \{\text{depth of } \mathbf{x}\}.$$

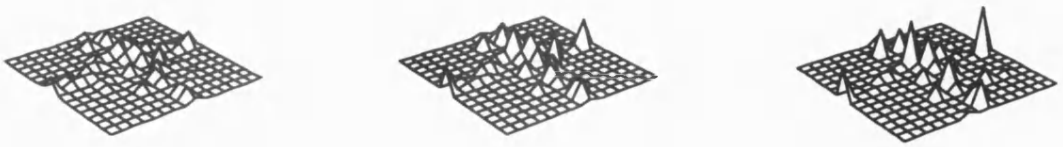
The depth of a local minimum was defined in Section 2.3.1, to be the minimum net increase in energy required from a local minimum to be able to reach any image of lower energy. The convergence conditions are certainly satisfied when $T_k = c/\log(1+k)$, provided that the constant $c \geq d^*$. We will concentrate on Metropolis based simulated annealing because Hajek's result is necessary and sufficient; the Gemans' result for the Gibbs sampler is only sufficient.

In our extended test lattice, there are 16^2 interior nodes, each of which has exactly four neighbours. In this situation, it is feasible to use a recursive routine to calculate the optimal route from each local minimum to any point of lower energy. Having found this value for all local minima, we can find the maximum, the constant d^*, for the two test functions. We can then consider various logarithmic schedules taking c equal to, greater than, or less than d^*. We have chosen to use schedules with $c=2d^*$, $c=d^*$, and $c=d^*/2$. In the image context, we cannot find the constant d^*, and so an assessment of this type is not feasible. Obviously it is still not possible to consider monitoring over an infinite number of annealing steps. In the lattice, we have only 256 potential global minimisers, and an exhaustive search could be completed in that number of evaluations. We have chosen to truncate the logarithmic schedules after 10^6 steps, which in comparison to 256 might be considered "asymptotic". Each schedule is then instantaneously frozen (temperature zero annealing) by applying ICM to convergence.

We will first consider applying simulated annealing to the smooth lattice function, shown in Figure 6.4(a). Throughout as an initial distribution, we have used a distribution which is uniform across the nodes, that is we would select any node with equal probability. As stated, we will consider three logarithmic schedules, the first (A) has $c=2d^*$, the second (B) $c=d^*$, and the third (C) $c=d^*/2$; for this function, $d^*=0.0659$. By Hajek's result, only the first two schedules should attain the desired asymptotic convergence, the third schedule is too cold. As discussed in Section 6.2.4, the quantities monitored are the probability that the global minimum has been selected at some stage, the probability that it is the current selection, and the expected energy under the current sampling distribution. Since the test function lies on [0,1], the range of the expected energy is also [0,1]; under the initial uniform sampling distribution, the expected energy is 0.15.

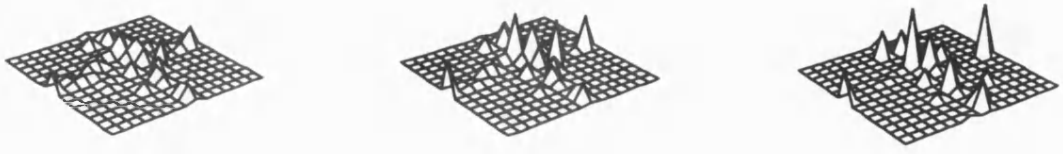
Figures 6.5 and 6.6, respectively, show perspective plots of the sampling distribution after various numbers of sweeps, and graphs of the monitored quantities. The perspective plots represent the sampling distributions after 10^3, 10^4, 10^5 and 10^6 sweeps of the three tested schedules. The maximum height of any peak is 0.2, so no node has a probability greater than 0.2 of being selected under the sampling distributions shown. We are expecting the $c=2d^*$ schedule to be too hot, and the $c=d^*/2$ schedule to be too cold. This appears to be the case. The $c=2d^*$ sampling distributions appear less focused on minima than their d^* and $d^*/2$ counterparts. The evolution with the number of sweeps of the sampling distribution is slow, but perceptible. Although, theoretically, convergence is guaranteed, there is insufficient pressure on the process to find low energy states. The final use of ICM forces the mass from its current position into the most accessible minima. For the $c=d^*/2$ schedule, the final application of ICM seems to have no effect. In fact, the distribution does not appear to be changing greatly even after as few as 10^3 steps. The mass has already been forced into the numerous local minima, and the schedule is too cold to allow much of it then to escape. The $c=d^*$ schedule appears to be a compromise; mass is finding local minima, but the schedule is not so cold that the distribution is already frozen. The improvement with the number of sweeps is noticeable, although it should be noted that the highest final peak does not correspond to $\mathbf{x}_{min}$; this peak actually corresponds to the node attaining the second lowest point of $V(\ )$, which has value 0.000163. This behaviour reflects the different basins of attraction for the different minima; this is particularly important when ICM is applied, because the process then looks for the path with one-step steepest descent.

The assessments based on the perspective plots are corroborated by the graphs of our three measured statistics, shown in Figure 6.6. One important point to note is the scale of the middle graph; despite having completed a very large



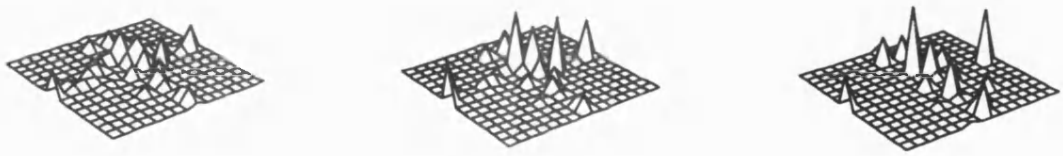
$c=2d^*$ $c=d^*$ $c=d^*/2$

Sampling distributions after 10^3 sweeps



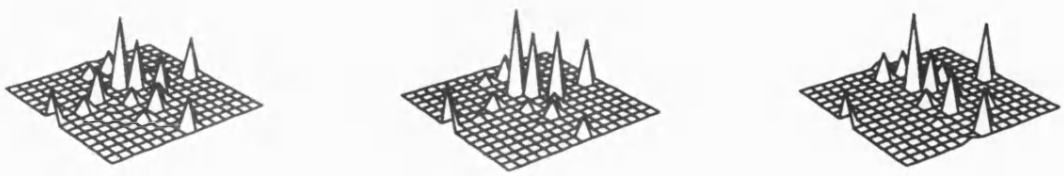
$c=2d^*$ $c=d^*$ $c=d^*/2$

Sampling distributions after 10^4 sweeps



$c=2d^*$ $c=d^*$ $c=d^*/2$

Sampling distributions after 10^5 sweeps



$c=2d^*$ $c=d^*$ $c=d^*/2$

Sampling distributions after 10^6 sweeps + ICM to convergence (maximum height 0.2)

Figure 6.5 The sampling distributions on the smooth function after the given number of sweeps of logarithmic schedules, and with the stated values of c.

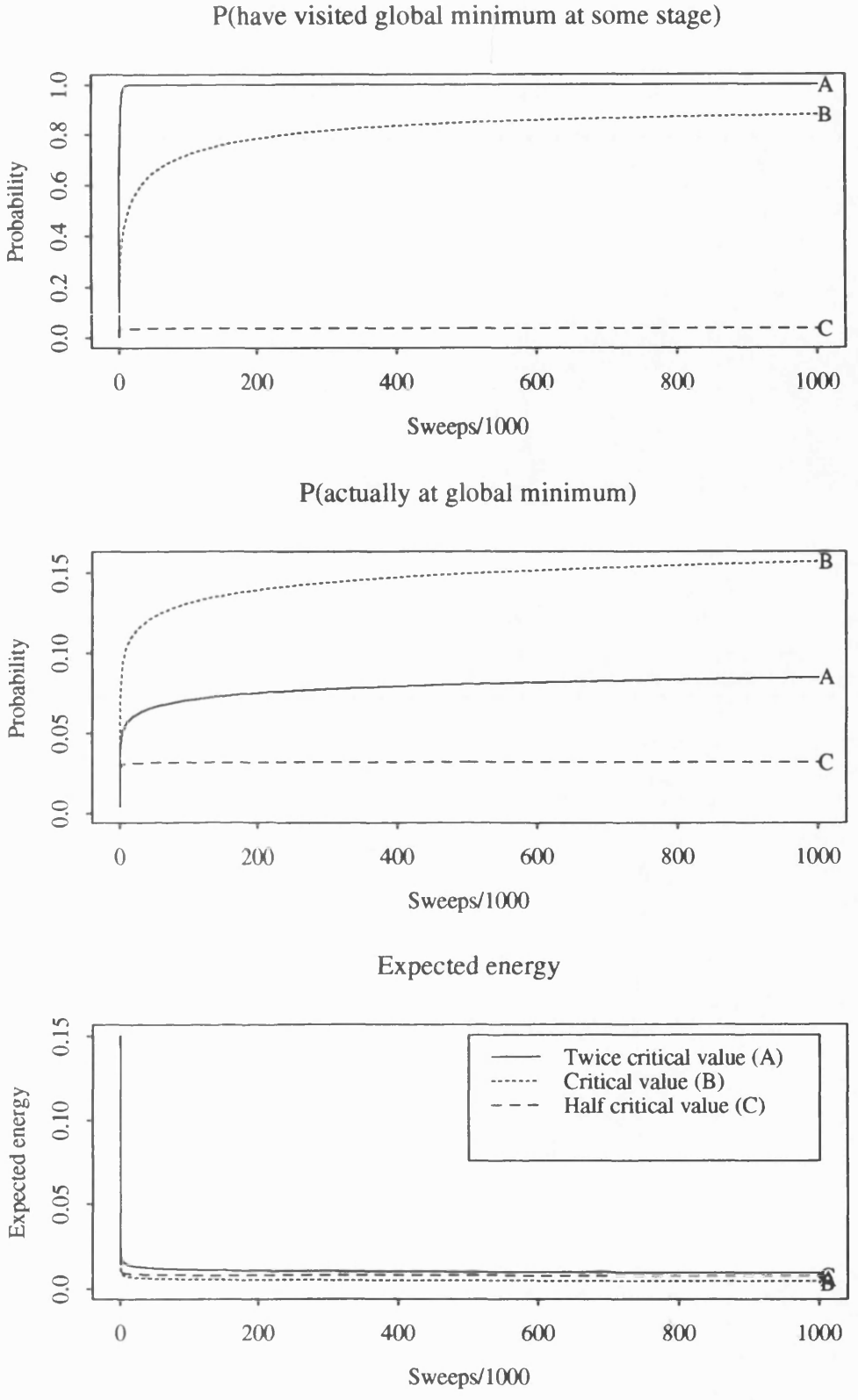
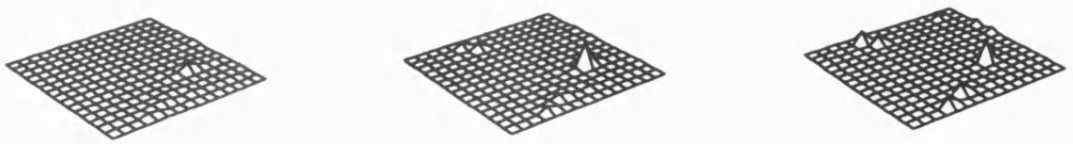


Figure 6.6 Results from 10^6 sweeps of logarithmic schedule with varying c for smooth test function.

number of sweeps, we have at best a probability of 0.157 of sampling the global minimum on the final sweep. This value is attained by the d^* schedule; this schedule also has the highest probability of sampling the global minimum, and the lowest expected energy at all stages. The probability of ever having selected the global minimum is clearly maximised by the $2d^*$ schedule. In some ways, this statistic is a measure of whether a schedule is too hot; the $2d^*$ schedule is behaving too randomly, it is wandering around the lattice too freely, and so is almost certain to visit $\mathbf{x}_{min}$ at some stage. For the $d^*/2$ schedule, the corresponding probabilities are only marginally greater than the probabilities of actually selecting the global minimum at any particular step. Essentially, we have remained in whatever minimum we reached in the first few steps of the simulated annealing. Although the final probability of selecting the global minimum is not high, the expected energy falls rapidly to a low value for all three schedules. This might be expected from considering the smooth function $V(\ )$ which is essentially a noisy quadratic. The structure allows mass to move rapidly down into the central basin where it is faced with a large number of similar low-energy minima. The $d^*/2$ schedule becomes trapped in these, while the other two schedules find it hard to escape to, or remain in, the global minimum. This problem is particularly difficult since the depths of many of the local minima are comparable with the energy rise which would be sufficient to escape from the global minimum.

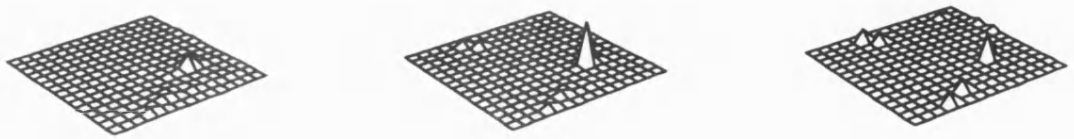
Figures 6.7 and 6.8 shows the corresponding perspective plots and graphs for the rough test function, Figure 6.4(b). In this case, $d^*=0.4$ exactly, and the expected energy under a uniform initial distribution is 0.522. The perspective plots in Figure 6.7 suggest that this is an easier problem to tackle than the minimisation of the smooth test function. There are fewer local minima, and these are better separated with larger basins of attraction. The differences between the behaviour of the three schedules are similar to, if not more pronounced than, those exhibited in the first example. In this case, the largest peak in all the plots corresponds to $\mathbf{x}_{min}$. The $d^*/2$ schedule traps some mass on the wrong side of the central high-energy region of $V(\ )$, and this does not seem able to escape. The sampling distribution appears to be frozen after the first 10^3 steps. In the d^* schedule, there is also some mass in this position, however the probability of sampling $\mathbf{x}_{min}$ is clearly increasing with the number of sweeps. Using the $2d^*$ schedule, the mass appears to have focused generally over the correct side of the high region, but the instantaneous freezing then forces the mass into some of the wrong minima on this side, demonstrating that it has not yet clearly determined the global minimum.

Again the corresponding graphs, shown in Figure 6.8, appear to match our interpretations from the perspective plots. The scale of the probability of sampling $\mathbf{x}_{min}$ at step k shows that we are much more successful with this problem than we



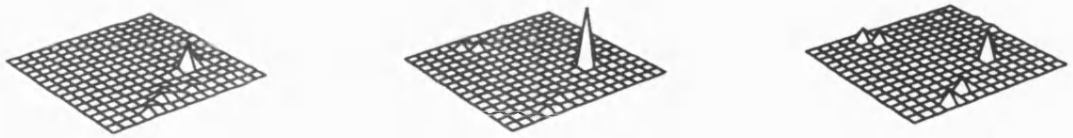
$c=2d^*$ $c=d^*$ $c=d^*/2$

Sampling distributions after 10^3 sweeps



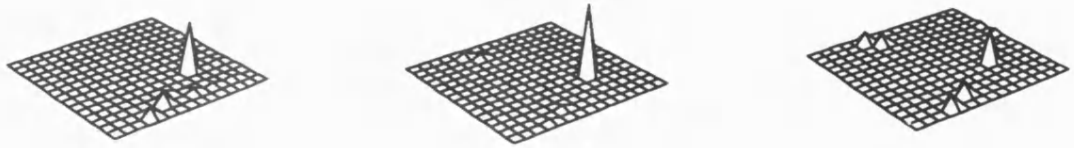
$c=2d^*$ $c=d^*$ $c=d^*/2$

Sampling distributions after 10^4 sweeps



$c=2d^*$ $c=d^*$ $c=d^*/2$

Sampling distributions after 10^5 sweeps



$c=2d^*$ $c=d^*$ $c=d^*/2$

Sampling distributions after 10^6 sweeps + ICM to convergence (maximum height 0.8)

Figure 6.7 The sampling distributions on the rough function after the given number of sweeps of logarithmic schedules, and with the stated values of c.

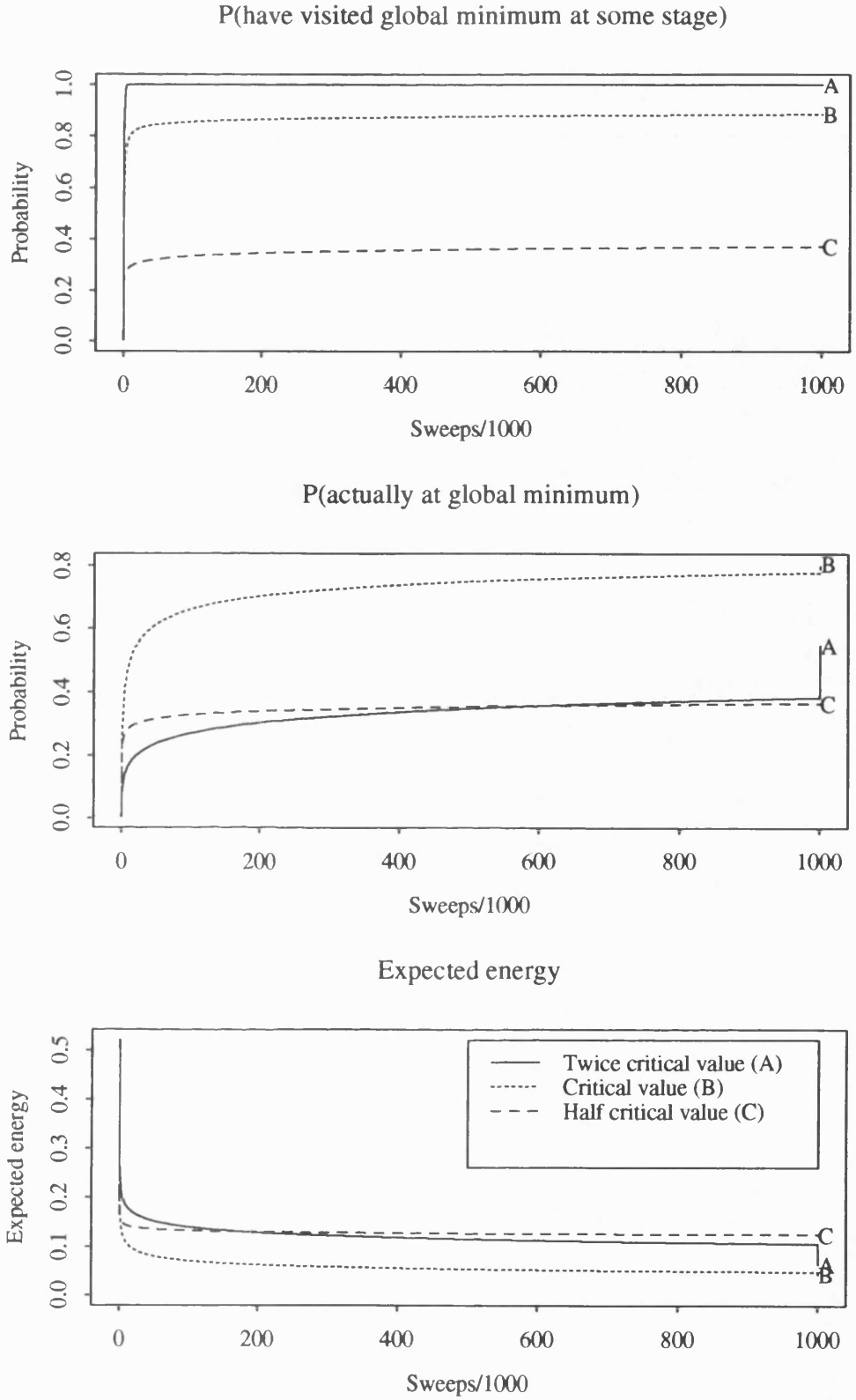


Figure 6.8 Results from 10^6 sweeps of logarithmic schedule with varying c for rough test function.

were with the previous problem. It also seems that having sampled the global minimum we are less likely to escape again, except for the $2d^*$ schedule which is wandering around too freely. However, the decreases in the expected energy are less dramatic, demonstrating that here the local minima are not comparable in value to the global minimum. In this case, the effect of ICM is noticeable with the d^* schedule and $2d^*$ schedule, in particular.

The differences in the behaviour of the two problems, and the related scales of d^*, reflect the geometrical dissimilarities between the two test $V(\ )$ functions. Independently of whether we wish to accept the image analogy, these results certainly show that the convergence of the sampling distribution to the target distribution could not be considered as fast or satisfactory.

6.3.2 Speeding up the logarithmic schedule

In Section 2.3.2, we demonstrated that logarithmic schedules of the type considered in the last section were not the only logarithmic schedules which satisfied Hajek's conditions. We suggested modifying the schedules to be of the theoretically acceptable form $T_k=d^*/\log(n+m(k-1))$ for finite, positive integers m and n. In the usual form, $n=2$; increasing n, for fixed m, is equivalent to starting the schedule at $k=n+1$ rather than at $k=1$. Similarly, in the usual schedule, $m=1$; increasing m, for fixed n, is equivalent to accelerating the schedule by only using every m^{th} value. In this section, we will present a few examples of these types of modification, in order to see whether they might be worth adopting.

We will first consider accelerating the standard logarithmic schedule, by increasing m for fixed $n=2$. Figure 6.9 shows the results from 10^6 steps, on the rough function, using the schedules $T_k=d^*/\log(2+m(k-1))$ for $m=1$, 5 and 10. Notice that the $m=1$ results are the standard logarithmic, and have already been plotted as the curves (B) in Figure 6.8. The results suggest that, in the long term for large k, there is no apparent benefit from accelerating the logarithmic schedule. However, the scale of the graphs may obscure any detail for the initial stages of the procedure, and so Figure 6.10 gives an enlarged view of the first 10^3 steps. The first plot, depicting the probability of ever selecting the global minimum, is not particularly informative, and has been replaced by a figure showing the comparative temperatures of the three schedules. This illustrates the effect of increasing m; the early fast cooling is more marked, while the tail, although cooler, is not perceptibly steeper.

On the increased scale of Figure 6.10, it can be seen that the faster cooling schedules do have an initial advantage. The probability of sampling the global minimum at step k is increased, while the corresponding expected energy is decreased. This advantage appears to be short term; although the $m=10$ schedule gives the highest probability of sampling the global minimum at the k^{th} step for

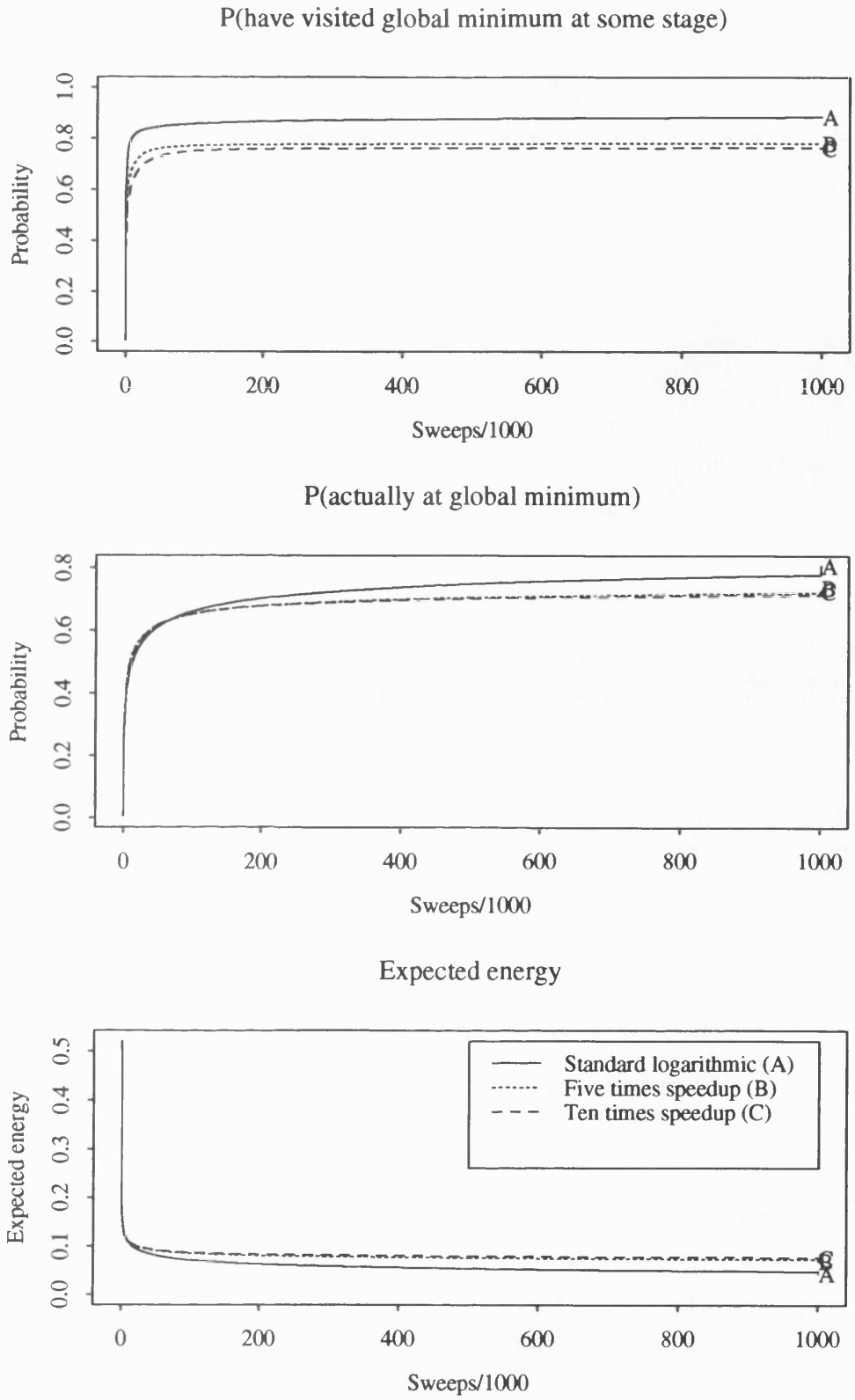


Figure 6.9 Results from 10^6 sweeps using accelerated logarithmic schedules with the constant $c=d^*$, and for rough test function.

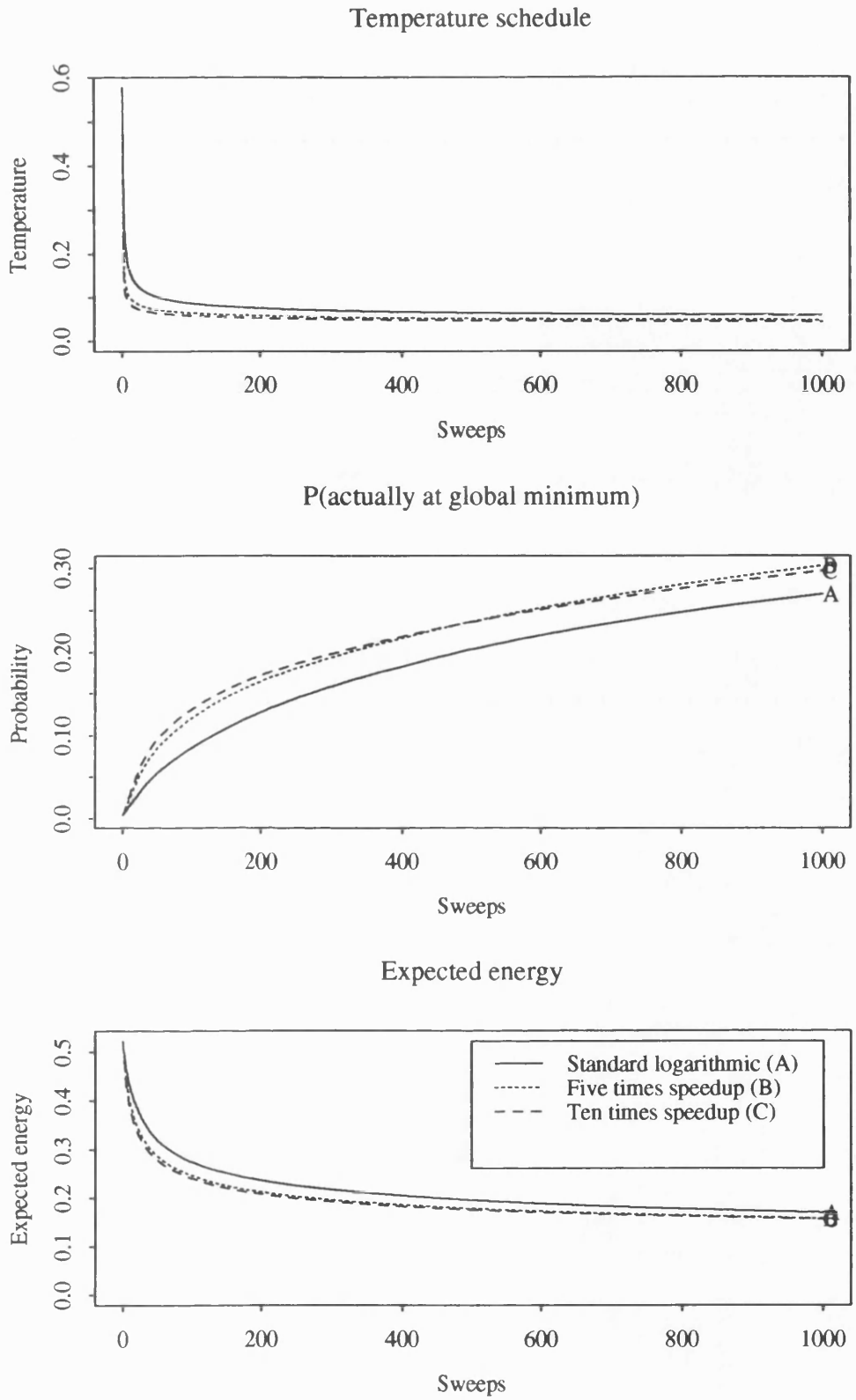


Figure 6.10 The enlarged view of the first 10^3 sweeps of Figure 6.9, replacing P(have visited global minimum at some stage) by the schedules.

the initial k, it is overtaken by the m=5 schedule around k=500. Similarly, from Figure 6.9, the m=5 schedule loses its advantage to the original m=1 at around step 6×10^4. Limited experiments seemed to suggest that increasing the value of m can increase the magnitude of this advantage, but also reduce its duration in terms of the number of steps; this may be relevant to annealing with fewer sweeps. The expected energy curves exhibit a similar, although less pronounced, behaviour with different cross-over times. In later examples, we will usually consider an accelerated schedule in addition to the standard logarithmic for comparison.

The effect of increasing n for fixed m is to start the schedule later, rather than to increase the rate of cooling. Various values of n were considered, ranging from around 10 through to values in the region of 10^4. It does not seem to be beneficial in this particular example, and no results are illustrated as the effects appeared to be minimal. In other examples, it might actually be the case that increasing n might have a detrimental effect. The first few relatively high temperature steps of the schedule will favour a uniform distribution over the nodes. This prevents the system settling too soon into local minima. In this case the initial distribution is already chosen to be uniform over the nodes, and so skipping the initial stages should not be a problem.

6.3.3 Annealing with a small number of sweeps

Although the computations in the last two sections are interesting in terms of assessing Hajek's theorem in action, this theorem is an asymptotic result. In practice, we would not carry out more steps of simulated annealing than the number of potential minimisers; almost certainly we would only wish to use a small fraction of this number. In this section, we will monitor the lattice sampling distributions over a much smaller number of sweeps, which we will refer to as finite-sweep annealing, as opposed to the 10^6 sweep situation which we considered as "infinite" for Hajek's result. Our test lattices consist of 256 nodes, and so in this discussion of finite-sweep schedules, we will limit ourselves to a maximum of 256 steps of simulated annealing. Already, in comparison to the imaging case where there are $N^{|S^0|}$ possible minimisers, this ratio is far in excess of what is computationally feasible.

In Section 2.3.2, we showed that for any strictly positive, decreasing schedule, defined for k=1, ..., K, there is a logarithmic schedule satisfying Hajek's conditions which is cooler for at least these K steps. Following the work of Stander (1992), which compares a number of different families of schedule, before recommending the use of a linear schedule, we will only present a comparison of three types of schedule. The first is the standard logarithmic $T_k^{(A)}=d^*/\log(1+k)$, for k=1, ..., $K-1$, followed by zero temperature annealing to convergence (ICM). The second is a linear schedule, chosen to start at the same initial value $d^*/\log(2)$,

and to decrease to 0 in K steps, $T_k^{(B)}=d^*/\log(2)\times(K-k)/(K-1)$. The third is another logarithmic schedule finished with ICM, again with the same starting value, but incorporating an acceleration factor as discussed in Section 6.3.2. We have chosen the acceleration so that this schedule is cooler than the whole of the first logarithmic schedule for $k\geq 2$, by setting the second value of this schedule equal to the $(K+1)^{th}$ value of the original logarithmic, $T_k^{(C)}=d^*/\log(2+K(k-1))$.

The three schedules, $T_k^{(A)}$, $T_k^{(B)}$, $T_k^{(C)}$, are applied with $K=256$ for our two test functions. The results for the smooth and rough functions are shown in Figures 6.11 and 6.12, respectively. These figures illustrate the schedules, the probability of sampling the global minimum at step k, and the expected energy at step k. As noted in Section 6.3.1, the two test functions produce graphs on different scales. The probability of sampling $\mathbf{x}_{\min}$ at the end of the annealing is less than 0.06 for the smooth function, and less than 0.3 for the rough function. There is a similar scale difference for the expected energies. Although the scales and curve shapes differ, the final orderings of the statistics over the three schedules are the same for the two functions. The standard logarithmic gives the highest final probability of selecting the global minimum, and the lowest expected energy at $k=256$. The benefit in completing this standard logarithmic schedule with ICM is clear. The linear schedule produces the poorest behaviour while it is hotter than the standard logarithmic, but noticeably improves as it becomes cooler towards K steps. The accelerated logarithmic is initially the best option, but gives the worst values at the final zero temperature step. Experiments with these three types of schedule truncated after various smaller values of K produced a similar ordering; the standard logarithmic gave the best final result, while the rating of the other two schedules alternated.

When the schedules are to be followed by ICM, it seems that accelerating the logarithmic schedule may not be beneficial, at least for the value we have used, namely $m=K$. The linear schedules perform somewhat better. However, considering all the steps of the simulated annealing in Figures 6.11 and 6.12, it seems that the linear schedules only begin to become attractive at larger k; for the early stages, the linear schedule produces poor values of the statistics we have chosen to measure. This does not seem surprising upon consideration of the relative temperatures of the linear and the logarithmic; the linear schedule is too hot most of the time, the sampling distribution is not concentrating on the good minima. The performance only improves when the temperature of the linear schedule is comparable to that of the logarithmic. The experiments with small values of K suggest that the differences in performance decrease with earlier truncation. In these situations, the linear schedule is cooler than the logarithmic for a larger proportion of the total number of steps. Suppose we consider how long

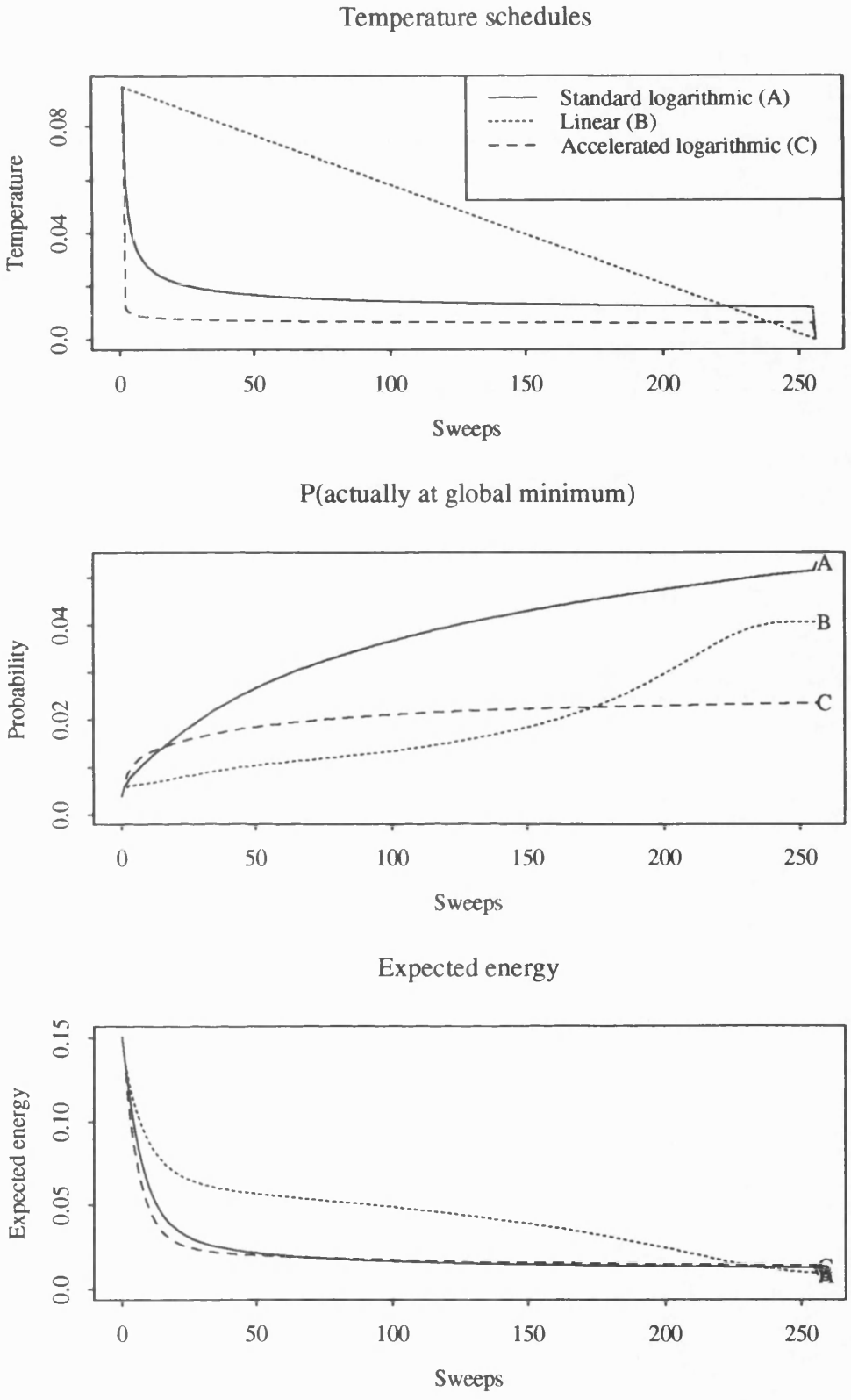


Figure 6.11 A comparison of various properties over 256 steps of simulated annealing for the smooth function.

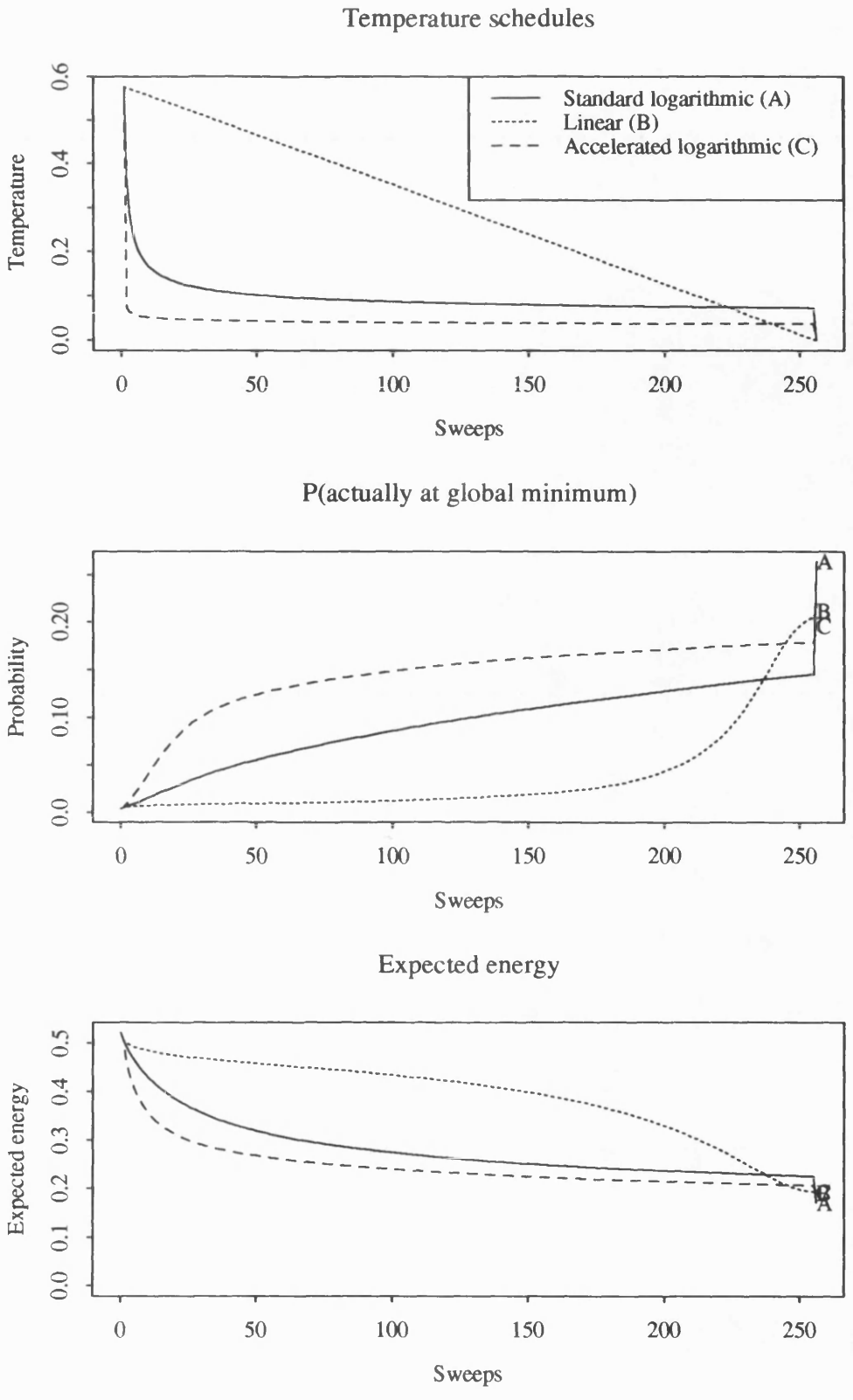


Figure 6.12 A comparison of various properties over 256 steps of simulated annealing for the rough function.

the schedule must be, that is the value of K, to allow a linear schedule to be consistently cooler than the logarithmic, assuming they start at the same value. Obviously, for the $T_k^{(A)}$ we have considered, K must be very small to allow this, since for the first few steps of the schedule, the temperature is rapidly decreasing. If we were to start the schedule later, using $T_k^{(D)}=d^*/\log(n+k)$, then increasing n as considered in Section 6.3.2, the schedule would be flatter at all steps. Correspondingly, we could employ a shallower linear schedule, starting from the same value, but then cooler than the logarithmic. Although, in Section 6.3.2, we found that the late-started logarithmic schedules were not particularly useful for the full annealing process, we will consider them here as they will be used when incorporating cascade. To ensure that the linear schedule is cooler than the logarithmic for $k\geq 2$, it is sufficient to ensure that it takes a lower value at $k=2$,

$$T_{k=2}^{(D)} \geq T_{k=2}^{(B)} \qquad \text{for schedules of length } K$$

$$\Longleftrightarrow \qquad \frac{d^*}{\log(n+2)} \geq \frac{d^*}{\log(n+1)} \left(\frac{K-2}{K-1}\right)$$

$$\Longleftrightarrow \qquad K \leq \frac{\log\left((n+2)^2/(n+1)\right)}{\log\left((n+2)/(n+1)\right)}. \tag{6.8}$$

Figure 6.13 illustrates three pairs of logarithmic and linear schedules, applied to the rough function, and following the equality in Equation (6.8) with the values $n=1$, 4 and 14. As n increases, the logarithmic schedule is flatter, and K increases; Equation (6.8) gives the corresponding K as 3, 10 and 43. The three sets of graphs in Figure 6.13 are scaled horizontally to reflect the different schedule lengths. As usual, the graphs depict the schedules, the probability of selecting $\mathbf{x}_{\min}$ at step k, and the expected energy at step k. The comparative performance of the linear schedule is good, although the logarithmic schedule is slightly superior in all cases. The final values for the probability of sampling $\mathbf{x}_{\min}$, and the expected energy are given below in Table 6.1, to 3 decimal places.

	$n=1$		$n=4$		$n=14$	
P(sampling $\mathbf{x}_{\min}$ at $k=256$)	log:	0.130	log:	0.135	log:	0.165
	lin:	0.130	lin:	0.128	lin:	0.151
Expected energy at $k=256$	log:	0.290	log:	0.276	log:	0.232
	lin:	0.290	lin:	0.283	lin:	0.244

Table 6.1 Final values for the simulated annealing shown in Figure 6.13.

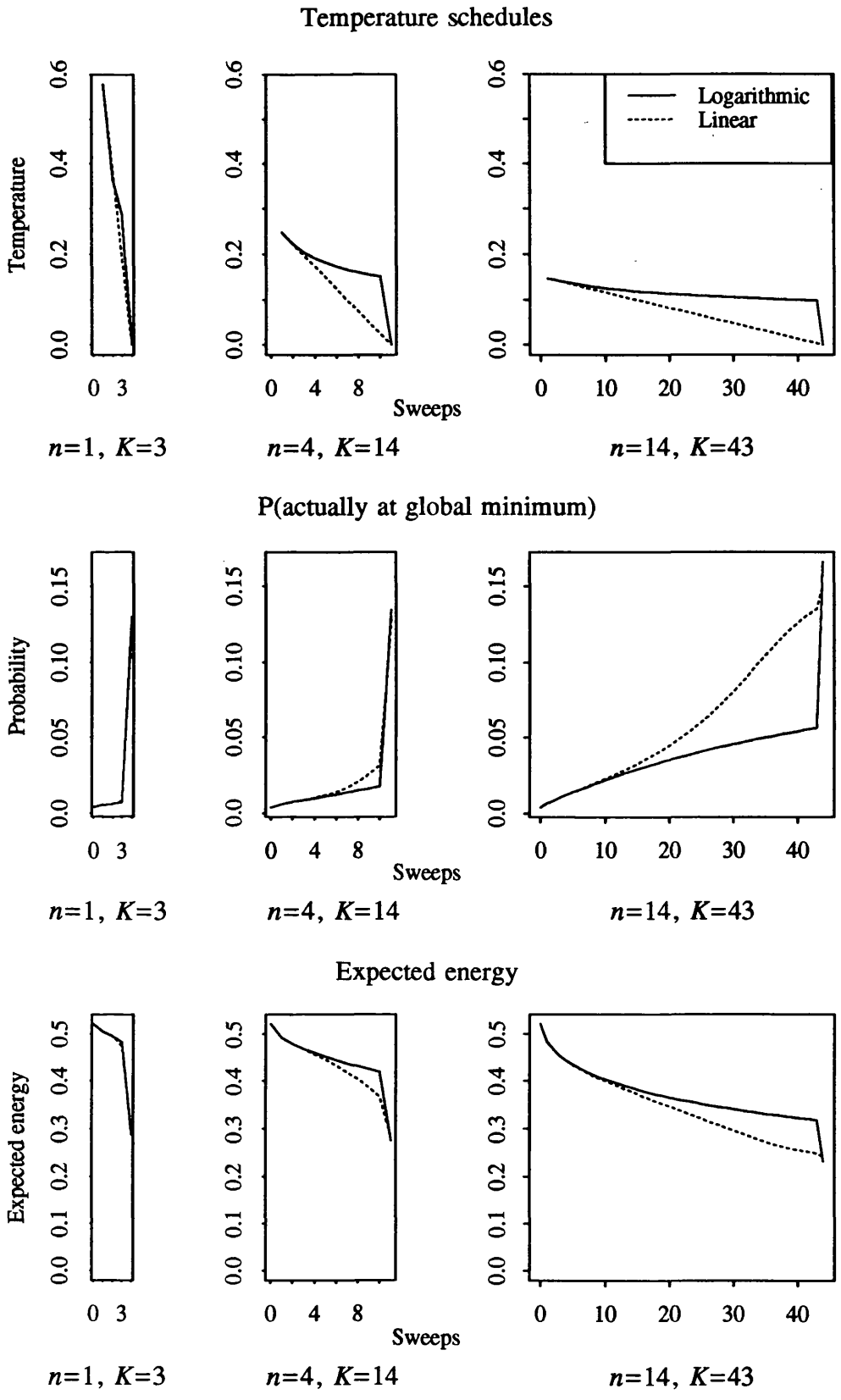


Figure 6.13 A comparison of logarithmic versus linear schedules for various delayed starts with the corresponding truncation rule.

6.4 Introducing cascade steps into the schedule

6.4.1 Temperature schedules

The main reason for defining the alternative lattice problem was to investigate the effect of introducing cascade steps into the simulated annealing. The choice of lattice deletion scheme for the cascade levels was discussed in Section 6.1, and the resulting reduced test functions were shown in Figure 6.4(c)-(f). In this section, we will consider various aspects of incorporating cascade steps into the annealing schedules, before presenting some examples using such schedules on our two test functions in the next section.

On our lattice we have four levels of the cascade. Following the notation of Chapter 4, we will denote the full 16×16 lattice by $L^{(0)}$, the 8×8 sublattice by $L^{(1)}$, and so on, up to $L^{(3)}$ in this case. The lattice levels give rise to four separate minimisation problems. For each $L^{(i)}$, we can find Hajek's $d^{*(i)}$, and anneal following a schedule which, asymptotically, would find that level's global minimum $\mathbf{x}_{\min}^{(i)}$. On our two test examples, $\mathbf{x}_{\min}=\mathbf{x}_{\min}^{(0)}\neq\mathbf{x}_{\min}^{(i)}$, $i\geq 1$; each of these higher $\mathbf{x}_{\min}^{(i)}$ does not necessarily have zero depth in $L^{(i-1)}$, that is a strictly downhill search on $L^{(i-1)}$ starting at $\mathbf{x}_{\min}^{(i)}$ may not be able to locate $\mathbf{x}_{\min}^{(i-1)}$.

Suppose we begin with any initial distribution on $L^{(3)}$, and carry out n steps of annealing using the schedule $T_k^{(3)}=d^{*(3)}/\log(1+k)$ on the restriction of $V(\,)$ to this sublattice. It is then possible to pass the $L^{(3)}$ n−step sampling distribution as the initial distribution on $L^{(2)}$ by augmenting the distribution with zeros on the nodes in $L^{(2)}\setminus L^{(3)}$. We can carry out a further n steps of annealing on $V(\,)$, now restricted to $L^{(2)}$, using the schedule $T_k^{(2)}=d^{*(2)}/\log(1+k)$. This process can be repeated until we carry out a final n steps of annealing on the complete lattice using $T_k^{(0)}=d^{*(0)}/\log(1+k)$. Asymptotically, by Hajek's result, this process will find the global minimum $\mathbf{x}_{\min}$, as well as each higher $\mathbf{x}_{\min}^{(i)}$, since the initial distribution for each $L^{(i)}$ annealing will not influence its step n sampling distribution in the limit as $n\rightarrow\infty$. So, even if a higher $\mathbf{x}_{\min}^{(i)}$ is badly positioned in order to reach $\mathbf{x}_{\min}$, asymptotically the situation is no worse than it was without cascade. This is reassuring, but does not offer any incentive to include cascade steps. We must consider finite-sweep annealing, and determine whether cascade can improve the rate of convergence of the $L^{(0)}$ sampling distribution by providing a good initial distribution. We will attempt to investigate this question by means of simulations both on the lattice, and in Section 6.5 with images. We have not attempted an analytic treatment of the effect of cascade on the convergence rate. This would be desirable, and could be the subject for further research if our simulation results suggest that cascade is worth considering in conjunction with simulated annealing.

In this section with the lattice experiments, we are interested in comparing the finite-sweep performance of the minimisation with, and without cascade. As before, we will restrict ourselves to at most 256 steps of annealing. These steps must be divided in some way between the different levels. It seems reasonable not to allocate the steps equally; the dimension of the four successive minimisation problems increases from $L^{(3)}$ to $L^{(0)}$, and we may be able to obtain comparable performance on all levels while using fewer sweeps on the smaller problems. Also, although the sequence of minimisations is trying to find $\mathbf{x}_{\min}^{(3)}$, $\mathbf{x}_{\min}^{(2)}$, $\mathbf{x}_{\min}^{(1)}$ and $\mathbf{x}_{\min}^{(0)}$ in turn, we are mainly interested in the solution to the final $L^{(0)}$ problem; without knowledge of the connection between the successive minima, we may be prepared to accept less precision on the sublattices. We have chosen to divide the sweeps so that the total number of sweeps completed when we have annealed on $L^{(i)}$ is proportional to the number of nodes in $L^{(i)}$. The constant of proportionality is determined by the ratio of the total number of sweeps to the number of nodes in the full lattice $L^{(0)}$. In this case, since K equals the number of nodes, we have the step allocation given in Table 6.2. This pattern will be used for all the lattice experiments; we have not investigated finding optimal patterns.

Level	Number of nodes	Number of steps	Total number of steps
$L^{(3)}$	4	4	4
$L^{(2)}$	16	12	16
$L^{(1)}$	64	48	64
$L^{(0)}$	256	192	256

Table 6.2 Allocation of simulated annealing steps to the cascade levels.

Three types of cascade schedule will be used in comparison with a standard non-cascade logarithmic truncated by ICM, and denoted (A). The first of these, (B), is the sequence of $d^{*(i)}$ standard logarithmic schedules on the $L^{(i)}$ truncated by ICM after the allocated number of steps for that level. The second, (C), is identical to the first, except that the $L^{(2)}$, $L^{(1)}$ and $L^{(0)}$ schedules are later sections of the same logarithmic curves. The starting delay is selected to correspond to the number of steps already completed on higher sublattices. The final cascade schedule, (D), is the sequence of linear schedules beginning at the initial values of the late-start logarithmics. These linear schedules each satisfy Equation (6.8), and lie entirely below the corresponding late start logarithmics. All three schedules are illustrated in relation to the non-cascade logarithmic in Figure 6.14.

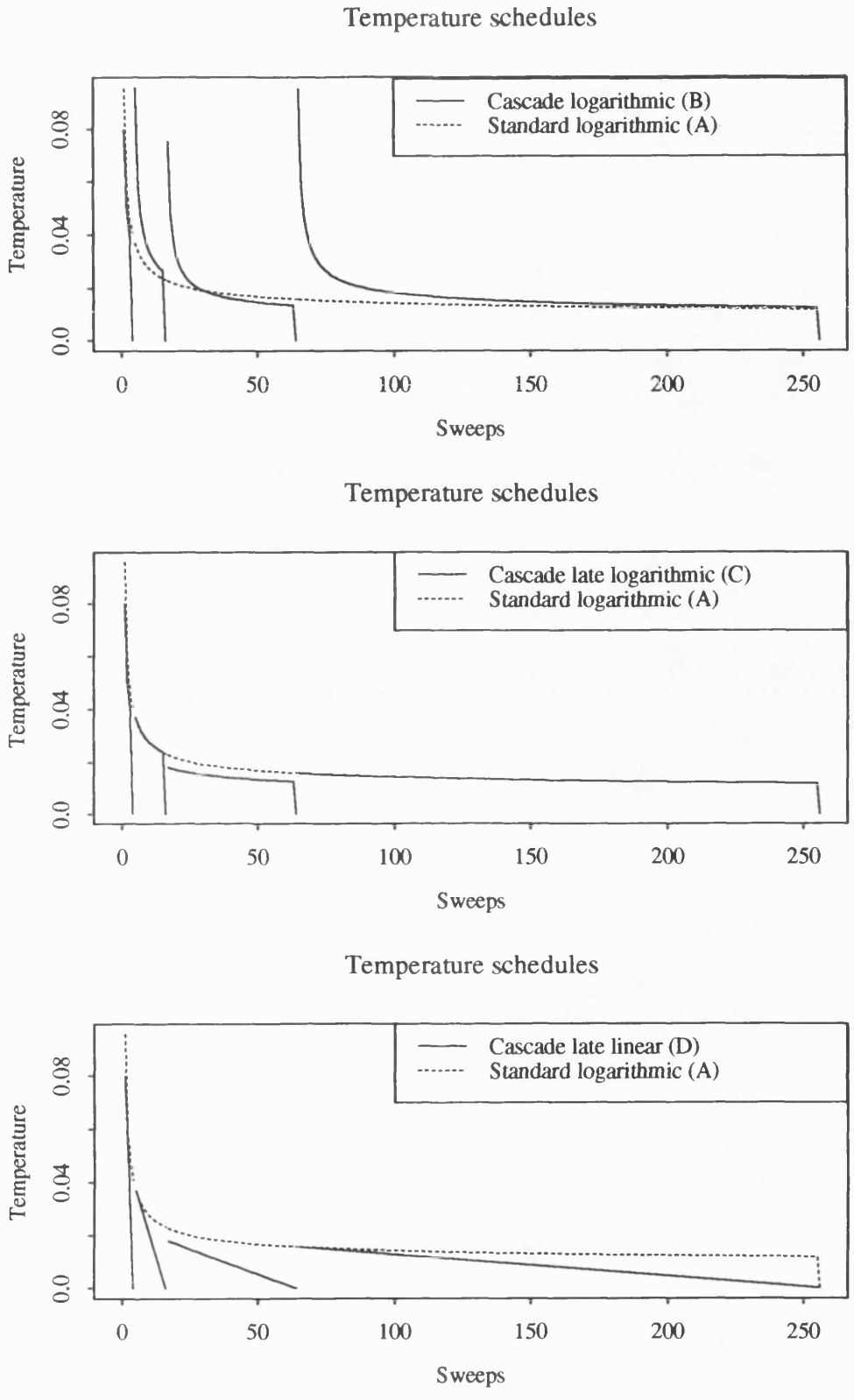


Figure 6.14 The three cascade schedules which will be used in comparison with the standard non-cascade truncated logarithmic.

6.4.2 Results

Figures 6.15 and 6.16 present the plots of the probability of sampling $\mathbf{x}_{min}$, and the expected energy for the smooth and rough test functions respectively, using the four schedules described in the last section. To clarify these plots, the final values of the probability of sampling $\mathbf{x}_{min}$, and the expected energy are given in Tables 6.3 and 6.4, to four decimal places.

Schedule	P(sampling $\mathbf{x}_{min}$)	Expected energy
Standard logarithmic (A)	0.0531	0.0082
Cascade logarithmic (B)	0.0398	0.0081
Cascade late logarithmic (C)	0.0249	0.0077
Cascade late linear (D)	0.0062	0.0063

Table 6.3 Final diagnostics for various schedules on the smooth test function.

Schedule	P(sampling $\mathbf{x}_{min}$)	Expected energy
Standard logarithmic (A)	0.2638	0.1708
Cascade logarithmic (B)	0.3121	0.1102
Cascade late logarithmic (C)	0.3221	0.0999
Cascade late linear (D)	0.3091	0.0905

Table 6.4 Final diagnostics for various schedules on the rough test function.

We will first discuss the results for the smooth test function, as given in Figure 6.15 and Table 6.3. The ordering of the success of the schedules, as measured by the final values of our two diagnostics, is exactly reversed between these two statistics. The schedules incorporating cascade produce lower expected energy distributions, but also have a lower final probability of actually sampling $\mathbf{x}_{min}$. It should be noted that these schedules must have a zero probability of sampling $\mathbf{x}_{min}$ before the final $L^{(0)}$ stage, since the global minimiser is not a node of any of the sublattices. Considering the plots in Figure 6.15, and also the functions to be minimised, Figures 6.4(a), (c) and (e), we can attempt to explain these results. The functions contain a number of minima taking similar low values, and situated in the central region of the lattice. The cascade schedules can reach certain non-deleted nodes of this region more rapidly than can the non-cascade

schedule, hence the rapid drop in expected energy. Successive cascade levels introduce more nodes, and the annealing attempts to allow investigation of these additional potential minimisers. However, as noted in Section 6.3.1, this test function is difficult to minimise as a result of its many similar, deep local minima. It seems that on passing from $L^{(i)}$ to $L^{(i-1)}$, the two late-start schedules can move out of certain local minima, incurring the initial increases in expected energy which can be seen in Figure 6.15. However, they are already sufficiently cold that they may be trapped within the deeper basins of low values, although quite successfully locating low energy values within these basins. This behaviour would be consistent with the observed low expected energies, but corresponding low probability of selecting $\mathbf{x}_{\min}$. The non-late-start schedule, being hotter, is more mobile and can explore further, hence the increased probability of sampling $\mathbf{x}_{\min}$, and the larger rise in expected energy on passing from one level to the next. Unfortunately, this schedule appears to suffer from the associated problem of being too hot with regard to finding a low final expected energy.

In comparison to the non-cascade schedule, the relative benefit of the schedules incorporating cascade depends on which of the two performance measures we consider more important. In Section 6.2.4, we argued that the expected energy of the sampling distribution was possibly more appropriate in the image case, since there, the best we can realistically hope is to select a good local minimum. Our speculative interpretation for the smooth test function was as a cascade-reduced portion of the full energy function. We have not demonstrated that the global minimiser of the cascade-reduced function is necessarily the best starting point for minimisation on the full energy function, if indeed this is the case. However, it does seem quite plausible that the lower energy states may be better starting values than the higher energy states.

Figure 6.16 and Table 6.4, the results for the rough test function, are rather more conclusive. In this case, the non-cascade schedule results in the worst value for both the expected energy, and the probability of sampling $\mathbf{x}_{\min}$. The three cascade schedules produce relatively similar values for the final probability of selecting the global minimiser. The late-start logarithmic actually produces the highest value of this diagnostic, although until ICM is applied, the late-start linear schedule appears to be by far the most successful. The two late-start schedules generate the lowest final expected energies. The rough test function, Figure 6.4(b), contains fewer, and more distinct, minima than the smooth test function. It seems, from the expected energy increases in moving from one cascade level to the next, that there is less need for an initial high temperature phase for these transitions. Indeed, the non-late-start schedule on $L^{(0)}$ appears to undo some of the work done on the higher levels.

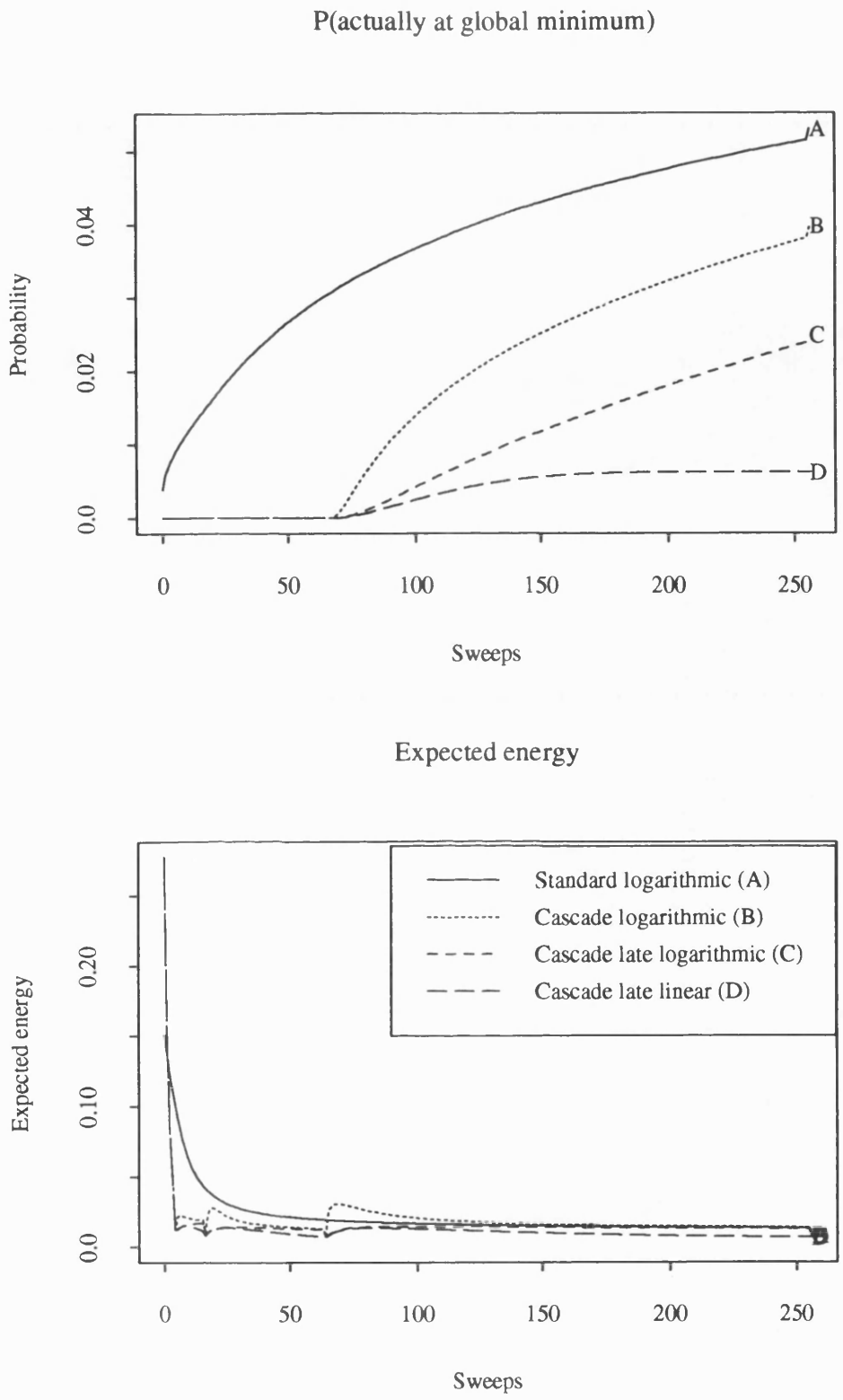
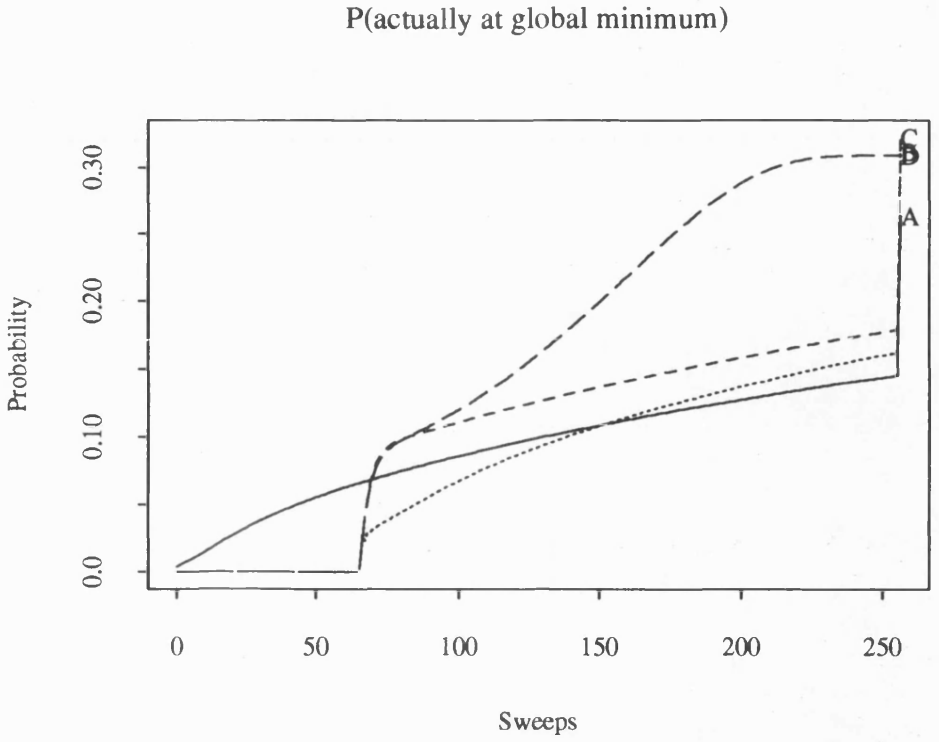


Figure 6.15 Some diagnostics for the smooth test function using the three cascade schedules and one non-cascade schedule.



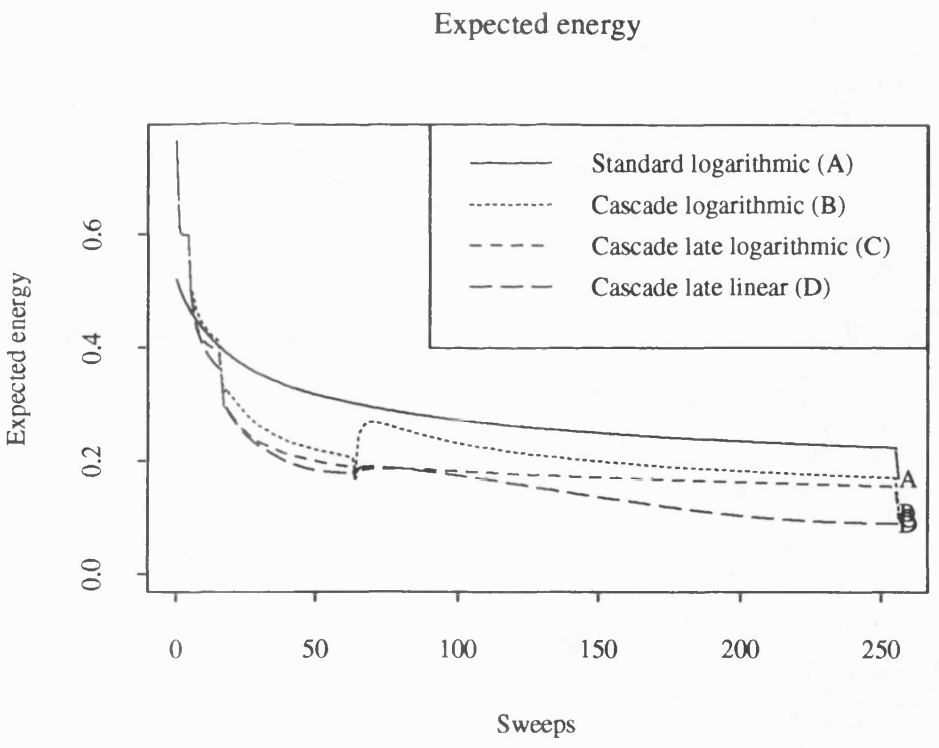


Figure 6.16 Some diagnostics for the rough test function using the three cascade scedules and one non-cascade schedule.

In Section 6.1.2, we conjectured that the rough test function could possibly have some of the characteristics of a low energy region of the complete energy function. If this is the case, then we might be equally interested in the two quantities we have chosen to measure. For this example, it does seem that introducing cascade steps into the schedule is beneficial for both the expected energy, and the probability of sampling the global minimum.

In conclusion from our lattice experiments, the addition of cascade does appear to be worth considering with simulated annealing. The primary benefit seems to be in finding a lower expected energy sampling distribution than non-cascade annealing finds in the same number of steps. The late-start schedules seem to be particularly effective. However, given the speculative nature of our image analogy, we must now consider the performance of cascade schedules in reconstructing images.

6.5 A return to the image problem

6.5.1 Temperature schedules

Returning to the image graph, we are in the situation where we have an extremely large number of nodes, $N^{|S^0|}$, the number of permitted grey-levels raised to the power of the number of pixels. We certainly cannot perform anywhere near this number of sweeps, particularly for reasonably sized grey-level images. Also as a result of the dimension of the problem, we cannot calculate the schedule parameter d^*, the maximum depth of the local minima. However, we need to select some family of schedules suitable for finite-sweep annealing including cascade, and we will do this in light of the results in the previous section.

A reasonable allocation of the annealing steps between the cascade levels is not as obvious as it was on the lattice. There it was possible to choose the number of steps at a particular level to be proportional to the number of additional nodes introduced at that level. When the total number of steps is equal to the total number of nodes, the step allocation is as shown in Table 6.2. In the imaging case, the number of images attainable at successive levels can grow rapidly, according to a factor equal to the number of grey-levels raised to the increase in the number of blocks. An allocation following lines similar to the lattice case would result in virtually all the steps being taken on the final full graph. Instead, we have chosen a sequence approximately proportional to the increase in the number of blocks at each level, rather than the number of attainable images. This is justified on two grounds; firstly it provides a reasonable number of steps at the higher levels. Second, it might be argued that the number of acceptable images, those with comparatively low energy, may be clustered in regions which grow more slowly in dimension than does the entire set of attainable images.

The schedules used for the simulated annealing examples in Chapter 5 were all one hundred sweeps in length. In order to produce comparable length schedules including cascade steps, we intend to divide the one hundred sweeps between the cascade levels according to the rough rule related to the number of blocks. Using a horizontal-vertical adaptive cascade, the number of blocks at successive levels increases by a factor of four. So following this rule, at the final level we would like to use roughly three quarters of the total number of sweeps. Three quarters of the remaining sweeps would then be used at the penultimate level, and so on. As all of the test examples are 64×64, and we have only 100 sweeps, following this rule would give the sequence {0, 1, 1, 5, 18, 75} (0 sweeps on the grid partitioned into 4 blocks, 1 sweep on the grid divided into 16 blocks, and so on). This scheme possibly has too few sweeps at the highest levels, and so we have decided to reallocate some sweeps while maintaining the total number; the sequence finally used is {1, 1, 2, 4, 24, 68}. Again, we have not investigated optimal sequences.

In implementing simulated annealing without cascade, we have used linear schedules decreasing from temperature 1, effectively sampling from the posterior at its natural temperature, to temperature 0, or ICM. In Section 6.3.3, such schedules did not compare particularly well with logarithmic schedules in the non-cascade experiments on the lattice. However, when cascade steps were introduced, their relative performance was much improved. As a result, we propose to use schedules of the type late-start linear described in the previous section (type (D)), and taking a common $c=\log(2)$ for all levels of the cascade. On each cascade level, we will follow the appropriate length linear schedule, decreasing to zero from the temperature the logarithmic would take if started after the number of steps completed on higher cascade levels.

There is a further implementation issue to be considered if we intend to use the window cascade described in Section 5.4.1. This window cascade was suggested in an attempt to improve the ICM adaptive cascade performance by using a more local partitioning scheme. Within the window, a complete cascade is performed, although the number of levels will clearly be less than in a complete grid cascade. This suggests that we should follow a similar argument for the allocation of sweeps between the window cascade levels, that is as a function of the number of blocks at each level. In order that the simulated annealing has sufficient sweeps to be able to find a reasonable minimum, we may need to use almost as many sweeps in total within the window, as for a complete grid cascade. Employing a window cascade may be quite computationally expensive, and resources would need to be redirected from the adaptive cascade to compensate for its introduction.

6.5.2 Examples

The images chosen as examples for this section are some of those considered in Chapter 5, two test scenes containing isolated objects, either with horizontal and vertical, or diagonal edges (records in Figures 5.4(b) and 5.9(b)), and the face image under four noise models (records in Figure 5.12). Previously these images were reconstructed using standard single-site ICM and simulated annealing, and also with various ICM implementations of cascade, including window cascades with a number of different partitioning models. In this section, we will make comparisons by using a simulated annealing version of cascade, permitted the same number of sweeps, and with the same initial temperature as the non-cascade annealing. Again we should point out that simulated annealing is a stochastic algorithm, we will only be presenting a single reconstruction from a single realisation of the process. Also it should be noted that we have implemented the Gibbs sampler for the image annealing, as opposed to the Metropolis algorithm used for the lattice experiments. As mentioned in Section 6.2.3, the Metropolis algorithm for grey level images will frequently propose values which stand little chance of being accepted. Although a revised version of the algorithm was proposed in that section, Hajek's theorem does not apply to this modification, and in addition we have not yet investigated this in practise.

We will begin by using simulated annealing with an adaptive horizontal-vertical cascade (abbreviated to HV), and following the schedule described in the previous section. That is, we will begin at temperature 1 on the highest cascade level, and at each level follow late-start linear schedules with the number of steps specified in the last section. The six reconstructions of the six records are shown in Figure 6.17. The model for Figures 6.17(a) and (b) is a four neighbour Geman and Reynolds' prior with $\Delta=10$ and $d=3$, and with $\xi=2$ for the adaptive cascade. The model for Figures 6.17(c)-(f) is an eight neighbour Geman and Reynolds' prior with $\Delta=10$ and $d=5$, and with $\xi=4$. No additional window cascades have been used in these reconstructions.

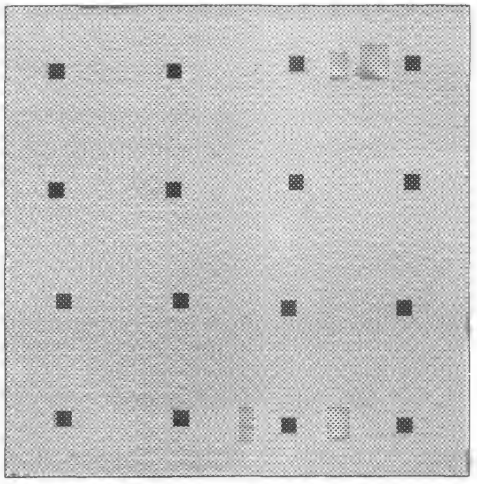
Figure 6.17 also states the final energy values for the reconstructions, and the time taken for the simulated annealing cascade as a multiple of that taken by single-site ICM to reconstruct the same record. Energy values for all six images, and timings for the latter four, have already been given for the other reconstruction methods in Chapter 5 (see Figures 5.13 to 5.16 for the timings). As might be expected, simulated annealing with cascade takes slightly less time than simulated annealing without cascade; the time spent in forming partitions is possibly recouped in processing smaller images at the higher levels. This time is also comparable to a series of window cascades using ICM, as detailed in Section 5.4. The six sets of energy values are summarised in Table 6.5.

Image	ICM	ICM HV cascade	ICM Window cascade	SA	SA HV cascade
(a)	487.2	451.7	433.3	436.6	**430.2**
(b)	504.2	462.0	452.3	462.5	**451.7**
(c)	1594.3	1576.3	1483.4	1524.8	**1471.4**
(d)	1300.4	1062.9	**964.8**	1093.4	1041.3
(e)	1331.7	1281.4	1105.5	1137.4	**1105.1**
(f)	1162.3	888.2	**844.8**	955.1	873.3

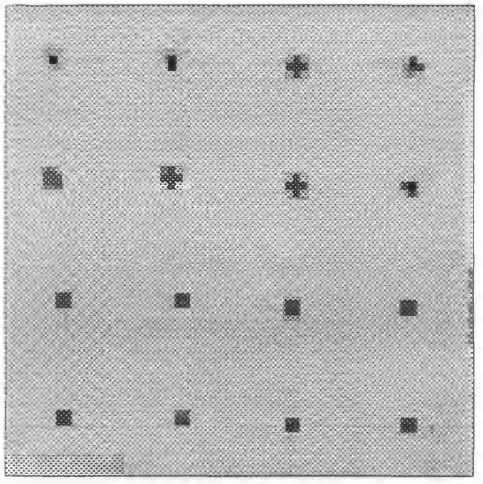
Table 6.5 Energy values after various reconstruction techniques for the six images, highlighting the lowest energy achieved for each.

In these test examples, the inclusion of cascade sweeps has consistently resulted in a lowering of the simulated annealing energy. The overall lowest energy for each image is produced either by simulated annealing with cascade (images (a), (b), (c) and (e)), or by the ICM window cascade (images (d) and (f), the two high noise records for the face image). In the cases where the cascade annealing does not achieve the lowest energy, it produces the second lowest value, outperforming the equivalent cascade with ICM.

In Figure 6.18, we have given the six plots of the energies of successive realisations generated by the two simulated annealing techniques, one with cascade steps, and the other without. There are marked similarities between the pairs of graphs for (a) and (b), (c) and (e), and (d) and (f). Images (a) and (b) are similar, and both are corrupted with the same form of the noise and blurring. The records for (c)-(f) are generated from the same scene, with either low or high noise, and either no or high blurring; records (c) and (e) share the same low noise levels, (d) and (f) the same high noise levels. Apart from the scale differences, the six curves for the simulated annealing without cascade are quite similar. There is an initial rise in energy, followed by a steady decrease subject to minor fluctuations. Towards the final sweeps, where the temperature is low, the graphs level out. The graphs for the simulated annealing with cascade are quite different, although caution must be taken in drawing too strong a conclusion from the differences, since the temperatures are not comparable at all steps. In (a), (b), (d) and (f), the energies of the cascade realisations begin at a lower level, and remain consistently lower than the non-cascade energies. There are clear increases in energy at the transitions between the levels, where the temperature is increased, and the

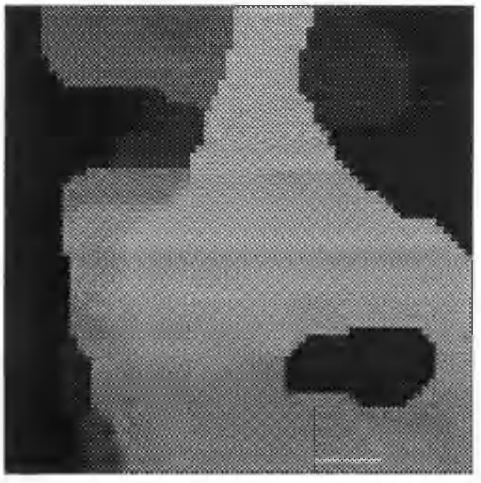


(a) Energy=430.2, 29×ICM time
Compare Figures 5.5(e), (f) and 5.7(f)

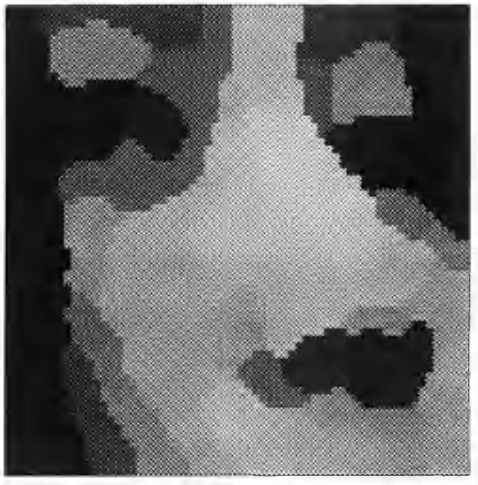


(b) Energy=451.7, 29×ICM time
Compare Figures 5.10(a)-(f)

(d) Energy=1471.4, 35×ICM time
Compare Figures 5.13(a)-(c)



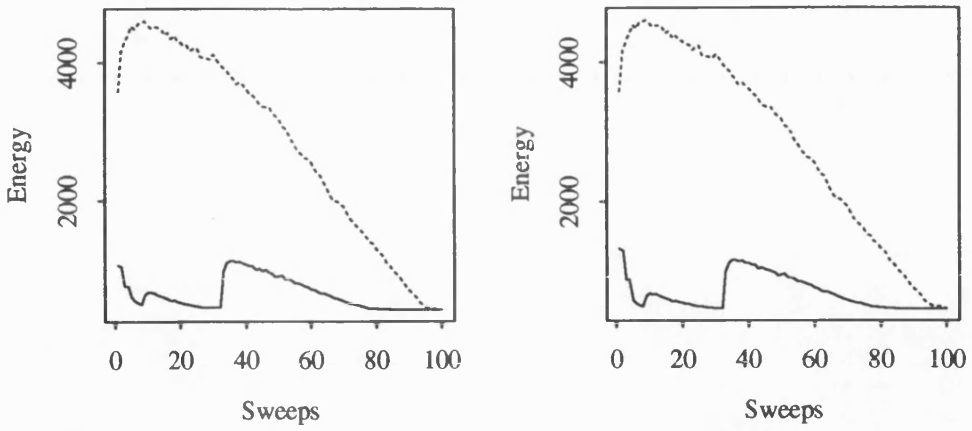
(c) Energy=1041.3, 32×ICM time
Compare Figures 5.14(a)-(c)



(e) Energy=1105.1, 11×ICM time
Compare Figures 5.15(a)-(c)

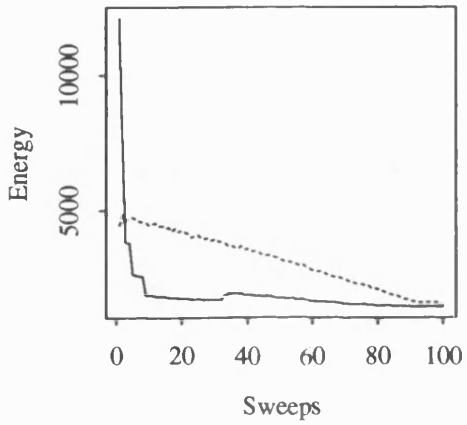
(f) Energy=873.3, 10×ICM time
Compare Figures 5.17(a)-(c)

Figure 6.17 Reconstructions after an HV adaptive simulated annealing cascade.

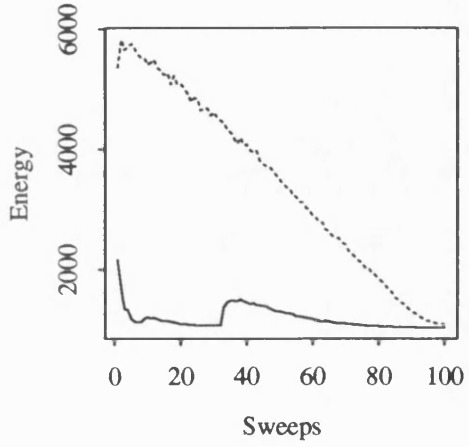


(a) Energy curves corresponding to reconstruction (a)

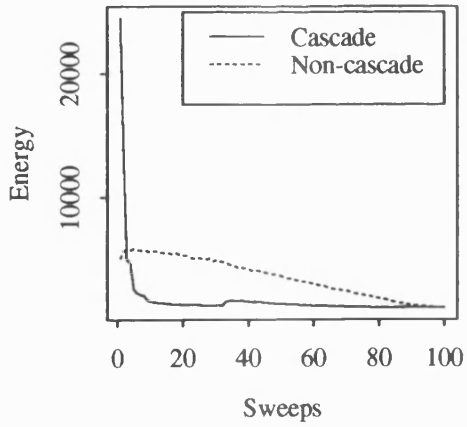
(b) Energy curves corresponding to reconstruction (b)



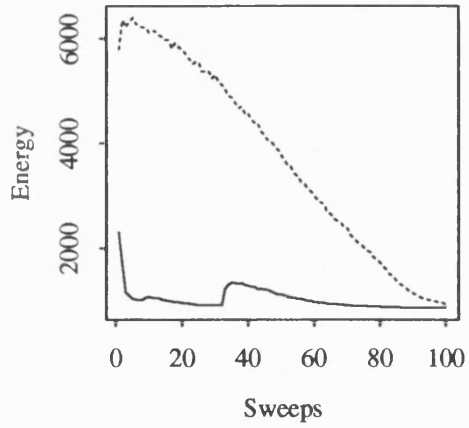
(c) Energy curves corresponding to reconstruction (c)



(d) Energy curves corresponding to reconstruction (d)



(e) Energy curves corresponding to reconstruction (e)



(f) Energy curves corresponding to reconstruction (f)

Figure 6.18 Energy of successsive realisations for the simulated annealing schedules.

blockings relaxed. The final sweeps of the annealing appear to be at a fairly gradually decreasing level. In the cases (c) and (e), the pattern for the cascade annealing is similar for the later sweeps, but there is a very high energy initial stage at the top of the cascade. In these two cases, the transition from one level to the next only results in an increase in energy for the final level. It is possible that these differences in features occur because these two records have the same low noise level. In this situation, the energy places greater weight on the data-fidelity contribution (this is apparent from Equation (3.24) which gives the formula for $\lambda=\lambda(\sigma)$, the constant balancing the competing factors). A high level cascade image will have low prior contribution, but high data-fidelity contribution unless the image is similar to (a) or (b), consisting largely of a uniform background. Although these graphs only represent the energies of particular simulated annealing realisations, it appears that the inclusion of cascade steps into the annealing is allowing the process to reach low energy states rapidly, without becoming trapped.

In Section 6.5.1, we mentioned combining simulated annealing with some form of window cascade, and noted that this procedure might be computationally expensive. The windows were initially introduced with ICM which, as a strictly downhill search technique, did not cope well when the partitioning was poor. By allowing a local partition, we were able to capture more difficult details without resorting to a more complex type of segmentation. Comparing Figure 6.17(a), where the HV adaptive cascade recovers most of the objects, to Figure 5.7(f), the reconstruction from the equivalent ICM HV adaptive cascade, simulated annealing appears to be less dependent on the partitioning. In this example, the monitoring of the energy of successive realisations showed an initial rise in passing from one level to the next. This suggests that the process may be able to escape from particularly bad regions if necessary, although this will obviously depend on the choice of relative starting temperatures for the different levels. If the aim in modifying simulated annealing is to produce the best performance given a fixed amount of computing resources, then these resources may be better utilised in allowing more sweeps for an adaptive cascade, than by sharing them between adaptive and window cascades. Limited experiments comparing these two approaches appeared to support this suggestion.

6.6 Conclusions

In this chapter, we have considered incorporating cascade steps into simulated annealing both on a small lattice, and also with images. Our choice of the small, manageable lattice, as an analogy for the image problem, enables us to find Hajek's constant, and to monitor exactly the sequence of generated sampling distributions with, and without, cascade. The lattice results suggest that, even with

the correct logarithmic schedule, simulated annealing can be extremely slow to converge. The long term behaviour appears to be heavily dependent upon the specification of the schedule, and the particular function to be minimised. The behaviour, in terms of finding the global minimum, is particularly bad when the function to be minimised contains a large number of similar minima.

We know that the introduction of cascade does not alter the existing asymptotic results for simulated annealing. Since asymptotic results are of little practical use, we have compared the finite-sweep performance of schedules with, and without, the cascade steps. Under our definition of cascade on the lattice, it seems that the introduction of the large steps is effective in finding low expected energy distributions, although the probability of actually sampling $\mathbf{x}_{min}$ can be reduced. It is possible that this cascade is allowing the process to find certain local minima quickly, while the temperature is still sufficiently hot to allow some escape to better minima. The influence on these results of our definition of a lattice cascade, that is the choice of sublattice, should be investigated, possibly by using a random deletion of nodes. Also, we need to consider how well our lattice analogy works for the image graph and energy.

In the image problem, we are restricted to monitoring the energy of a single realisation from each temperature sampling distribution. In experiments on images, incorporating schedule information from the lattice results, an adaptive horizontal-vertical cascade succeeded in producing lower energy solutions than standard simulated annealing, while expending roughly the same computational effort. The same improvement in performance was observed in Chapter 5 between standard single-site update ICM and a window cascade ICM, although in that case the CPU time required increased. Tracking the energy of successive realisations suggested that the cascade was generating low energy scenes more rapidly. The rate of convergence must depend upon the image graph underlying the energy, and the way in which cascade alters this graph. It seems that any theoretical work should be concentrated on an investigation of these areas. However, in these practical examples, cascade certainly does appear to be useful in directing the simulated annealing search towards a low energy solution.

Chapter 7: Conclusions and further work

In this thesis, we have considered a number of aspects of statistical image reconstruction. The work has fallen roughly into two areas: prior modelling of the scene, and possible improvements to existing reconstruction algorithms.

The reconstructions on which we have concentrated have been of grey-level images corrupted by both noise and blurring. In modelling these types of scene, certain features of the prior are desirable for the processing stages, both to remove the blurring, and also to recover discontinuities. We have discussed in detail the recent work of Geman & Reynolds (1992), which expresses these properties in a usable prior form. They also extend the usual type of smoothness constraints, which favour regions of constant grey-level, to higher order constraints which permit linearly, and quadratically smooth regions. Their corresponding three orders of model, working with the four nearest adjacent pixels as Markov random field neighbours, are tailored to recover horizontal and vertical discontinuities between regions of constant grey-level. In order to do this, the parameter of the energy function balancing smoothness with fidelity is specified as a function of the known noise and blurring. We have extended all three order models to include diagonally adjacent nearest pixels in the neighbourhood structure. We have then calculated the appropriate parameter setting so that diagonal first order discontinuities are also coordinate-wise minima of this revised prior, and should be recovered.

Experiments reconstructing heavily degraded images suggest that Geman and Reynolds' approach is effective. The hierarchical processing produces reconstructions which are more visually acceptable than those obtained using standard first order smoothness constraints alone. It also seems that it is beneficial to use the extended eight neighbour model in order to capture diagonal features. These experiments do reveal one unwanted artifact of working with a prior which has the finite asymptotic limit recommended for recovering discontinuities. We have identified a possible reason for these occurrences of extreme, outlying pixel values, and application of our recommended solutions appear to curb the problem. Geman and Reynolds do not suffer from this problem because they have employed a modified Gibbs sampler which restricts the potential pixel values. This algorithm is intended to reduce the computational burden of simulated annealing for grey-level images, but unfortunately it does not satisfy the conditions necessary for convergence of the sampler. We have stated the additional rejection step required to correct this problem, although adoption of this step may prevent the intended computational savings. A similar approach has then been applied to the Metropolis algorithm, making it more suitable for use with grey-level images, however this algorithm has not yet been implemented, and assessed.

Work still remains to be done with this particular modelling approach: The concept of a coordinate-wise minimum is linked to iterative algorithms which update one pixel at a time, particularly the standard implementation of ICM. A stronger condition would be that certain prototypical images have a high probability of being the global minimisers of their respective energy functions. This would require a much more complicated analysis. It also raises the question of how general a discontinuity could be dealt with by the model. One other remaining problem is the effect on the method of incomplete knowledge of the noise or blurring processes, and the use of estimates for these quantities.

Once the energy function has been defined, there are algorithmic problems to consider in attempting to locate the MAP estimate. A pixel-by-pixel application of ICM is unlikely to yield a particularly good local minimum. Simulated annealing employing some single-site update proposal mechanism should perform better in the long term, but theoretical results suggest that the cooling schedules must be very slow (although we have demonstrated that there is more flexibility in applying Hajek's theorem than is usually considered). We have presented a brief survey of some multiple site update methods which could be used in an attempt to improve the performance. Some of these approaches are inclined towards minimisation, and so could incorporate either ICM or simulated annealing. The remainder were originally conceived to improve sampling performance, and so are more relevant for incorporating into simulated annealing. The degree of theoretical complexity varies between the methods, as does the range of problems to which they are applicable, and the extent to which they are actually implementable. The efficient implementation of two of the methods described, the renormalised group approach, and the extended Swendsen-Wang algorithm are still research topics.

We have considered in depth the cascade algorithm, possibly the simplest multiple-site update method. This has been reformulated to counter some theoretical objections to its original form, and also extended to deal with blurred scenes. In this redefinition, recolouring blocks of pixels is equivalent to taking large steps around the section of the image energy function linked to a certain subgraph of all the possible images. The scheme for blocking pixels together determines the subgraph at each level of the cascade, and so is likely to be important in the performance. We have proposed, implemented and demonstrated various blocking schemes, which crudely attempt to identify and group pixels which may have similar behaviour. The results suggest that cascade can produce an improvement in the performance of ICM. However, with ICM there is still a fairly heavy dependence on the blocking scheme, and more work is needed to determine the interaction of the blocking scheme and the performance.

In order to investigate how cascade works, we have considered it in conjunction with simulated annealing. The prohibitive dimensions of the original problem led us to suggest an analogy with a minimisation problem defined on a smaller, regular lattice. The benefits of this alternative situation are that we can monitor the entire sampling distribution at a particular temperature, rather than just a particular realisation. It is also possible to find Hajek's constant required for the theoretical convergence, and so to assess "asymptotic" behaviour. As a result of these experiments, both with and without cascade, we made several observations. The first is that simulated annealing following a logarithmic schedule is strongly affected by misspecifying Hajek's constant. Even following a correct schedule, the convergence rate may be very slow, although this is more dependent on the function to be minimised; simulated annealing appears to struggle most when there are a large number of similar minima. There can be benefits from starting the logarithmic schedule late, or speeding it up, or even replacing it by a linear schedule, particularly when the number of sweeps is small. The effect of introducing cascade steps appears to be to produce a lower expected energy sampling distribution; on the lattice, cascade will help find a good local minimum.

Experiments using simulated annealing in conjunction with cascade on images, incorporating schedule information from the lattice problem, appear promising. A commonly used schedule is the linear. In our examples, a late-start linear cascade schedule was quite effective in comparison; in the same number of sweeps, and a slightly reduced amount of CPU time, the cascade version of simulated annealing produced lower energy solutions than the non-cascade version. As might be expected, simulated annealing is not as strongly affected by the choice of blocking as is ICM. Even when the higher level minima are not ideally located for the minima on the full graph, it seems that simulated annealing is still able to find low energy realisations.

We have not produced any theoretical results for the cascade algorithm. Such results would probably involve investigating the way in which the energy function is connected, and the effect on the problem of different choices of subgraph; they might be quite hard to obtain. However the idea of taking large directed steps around the energy function as a means of improving the performance of simulated annealing seems appealing. Cascade is certainly a readily understood, and easily implemented algorithm; it might be of interest to consider cascade equivalents, block updating of components with behaviour in some way similar, in other areas in which stochastic minimisation is required.

References

Besag, J. (1974). Spatial Interaction and the Statistical Analysis of Lattice Systems. *Journal of the Royal Statistical Society,* **B36**, 192-236.

Besag, J. (1986). On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society,* **B48**, 259-302.

Besag, J. (1989). Towards Bayesian Image Analysis. *Journal of Applied Statistics,* **16**, 395-407.

Besag, J. & Green, P. (1992). Spatial Statistics and Bayesian Computation. *Journal of the Royal Statistical Society,* **B55**, to appear.

Bouman, C. & Liu, B. (1991). Multiple Resolution Segmentation of Textured Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **PAMI-13**, 99-113.

Brandt, A., Ron, D. & Amit, D. (1985). Multi-level Approaches to Discrete-state and Stochastic Problems. In: *Multigrid Methods II, Lecture Notes in Mathematics* (editors Dold & Eckmann). Berlin: Springer-Verlag.

Briggs, W. (1987). *A Multigrid Tutorial.* Philadelphia: Society for Industrial and Applied Mathematics.

Cibulskis, J. & Dyer, C. (1984). Node Linking Strategies in Pyramids for Image Segmentation. In: *Multiresolution Image Processing and Analysis* (editor Rosenfeld). Berlin: Springer-Verlag.

Cross, G. & Jain, A. (1983). Markov Random Field Texture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **PAMI-5**, 25-39.

Edwards, R. & Sokal, A. (1988). Generalization of the Fortuin-Kasteleyn-Swendsen-Wang Representation and Monte Carlo Algorithm. *Physical Review,* **D38**, 2009-2012.

Geman, D. (1990). Random Fields and Inverse Problems in Imaging. *Lecture Notes in Mathematics.* Berlin: Springer-Verlag.

Geman, S. & Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **PAMI-6**, 721-741.

Geman, D., Geman, S., Graffigne, C. & Dong, P. (1990). Boundary Detection by Constrained Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **PAMI-12**, 609-628.

Geman, S. & Graffigne, C. (1986). Markov Random Field Image Models and their Applications to Computer Vision. *Proceedings of the International Congress of Mathematicians*, 1496-1517.

Geman, D. & Reynolds, G. (1992). Constrained Restoration and the Recovery of Discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **PAMI-14**, 367-383.

Gidas, B. (1989). A Renormalisation Group Approach to Image Processing Problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **PAMI-11**, 164-180.

Green, P. (1991). A Note on the Swendsen-Wang Algorithm and Ordered Colours. Technical Report, Department of Mathematics, University of Bristol.

Green, P. & Han, X. (1991). Metropolis Methods, Gaussian Proposals and Antithetic Variables. In: *Stochastic Models, Statistical Methods and Algorithms in Image Analysis* (editors Barone, Frigessi & Piccioni). Berlin: Springer-Verlag.

Hajek, B. (1988). Cooling Schedules for Optimal Annealing. *Mathematics of Operational Research,* **13**, 311-329.

Hastings, W. (1970). Monte Carlo Sampling Methods using Markov Chains and their Applications. *Biometrika,* **57**, 97-109.

Jubb, M. (1989). Image Reconstruction. Ph.D. thesis, School of Mathematical Sciences, University of Bath.

Jubb, M. & Jennison, C. (1991). Aggregation and Refinement in Binary Image Restoration. In: *Spatial Statistics and Imaging* (editor Possolo), 150-162. Haywood: Institute of Mathematical Sciences Lecture Notes.

Kandel, D., Domany, E. & Brandt, A. (1989). Simulations Without Critical Slowing Down: Ising and Three-state Potts Models. *Physical Review,* **B40**, 330-344.

Kirkpatrick, S., Gelatt, C. & Vecchi, M. (1983). Optimization by Simulated Annealing. *Science,* **22**, 671-680.

Marroquin, J., Mitter, S. & Poggio, T. (1987). Probabilistic Solution of Ill-Posed Problems in Computational Vision. *Journal of the American Statistical Association,* **82**, 76-89.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. & Teller, E. (1953). Equations of State Calculations by Fast Computing Machines. *Journal of Chemical Physics,* **21**, 1087-1092.

Molina, R. & Ripley, B. (1989). Using Spatial Models as Priors in Astronomical Image Analysis. *Journal of Applied Statistics,* **16**, 193-206.

Montanvert, A., Meer, P. & Rosenfeld, A. (1991). Hierarchical Image Analysis using Irregular Tessellations. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **PAMI-13**, 307-316.

Peskun, P. (1973). Optimum Monte-Carlo Sampling using Markov Chains. *Biometrika,* **60**, 607-612.

Ripley, B. (1988). *Statistical Inference for Spatial Processes.* (First paperback edition). Cambridge: Cambridge University Press.

Rosenfeld, A. (1984). Some Useful Properties of Pyramids. In: *Multiresolution Image Processing and Analysis* (editor Rosenfeld). Berlin: Springer-Verlag.

Silverman, B., Jennison, C., Stander, J. & Brown, T. (1990). The Specification of Edge Penalties for Regular and Irregular Pixel Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **PAMI-12**, 1017-1024.

Sokal, A. (1989). *Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms.* Lausanne: Cours de Troisieme Cycle de la Physique en Suisse Romande.

Stander, J. (1992). Some Topics in Statistical Image Analysis. Ph.D. thesis, School of Mathematical Sciences, University of Bath.

Swendsen, R. & Wang, J. (1987). Nonuniversal Critical Dynamics in Monte Carlo Simulations. *Physical Review Letters,* **58**, 86-88.

Switzer, P. (1983). Some Spatial Statistics for the Interpretation of Satellite Data. *Bulletin of the International Statistics Institute,* **50**, 962-972.

Terzopoulos, D. (1986). Image Analysis using Multigrid Relaxation Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **PAMI-8,** 129-139.

Thompson, A., Brown, J., Kay, J. & Titterington, D. (1991). A Study of Methods of Choosing the Smoothing Parameter in Image Restoration by Regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **PAMI-13**, 326-339.